

HOW SIMULATION CAN ILLUMINATE PEDAGOGICAL AND
SYSTEM DESIGN ISSUES IN DYNAMIC OPEN ENDED
LEARNING ENVIRONMENTS

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Stephanie Frost

©Stephanie Frost, January/2017. All rights reserved.

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

ABSTRACT

A Dynamic Open-Ended Learning Environment (DOELE) is a collection of learners and learning objects (LOs) that could be constantly changing. In DOELEs, learners need the support of Advanced Learning Technology (ALT), but most ALT is not designed to run in such environments. An architecture for designing advanced learning technology that is compatible with DOELEs is the ecological approach (EA). This thesis looks at how to test and develop ALT based on the EA, and argues that this process would benefit from the use of simulation.

The essential components of an EA-based simulation are: simulated learners, simulated LOs, and their simulated interactions. In this thesis the value of simulation is demonstrated with two experiments. The first experiment focuses on the pedagogical issue of peer impact, how learning is impacted by the performance of peers. By systematically varying the number and type of learners and LOs in a DOELE, the simulation uncovers behaviours that would otherwise go unseen. The second experiment shows how to validate and tune a new instructional planner built on the EA, the Collaborative Filtering based on Learning Sequences planner (CFLS). When the CFLS planner is configured appropriately, simulated learners achieve higher performance measurements than those learners using the baseline planners.

Simulation results lead to predictions that ultimately need to be proven in the real world, but even without real world validation such predictions can be useful to researchers to inform the ALT system design process. This thesis work shows that it is not necessary to model all the details of the real world to come to a better understanding of a pedagogical issue such as peer impact. And, simulation allowed for the design of the first known instructional planner to be based on usage data, the CFLS planner. The use of simulation for the design of EA-based systems opens new possibilities for instructional planning without knowledge engineering. Such systems can find niche learning paths that may have never been thought of by a human designer. By exploring pedagogical and ALT system design issues for DOELEs, this thesis shows that simulation is a valuable addition to the toolkit for ALT researchers.

ACKNOWLEDGEMENTS

Thank you to my supervisor, Dr. Gord McCalla, for teaching me how to be a researcher. Early on you explained techniques for reading papers and following references, and in later years coached me to evolve my writing beyond technical descriptions to speaking directly to the deep issues in AIED that fascinate me so. With this research area being so new, we didn't have much of a basis for comparison. Sometimes I'd lose confidence when my simulation output was different than expected. I always appreciated your insights, ideas and guidance toward a research problem that was manageable yet still an exciting and useful contribution.

Thanks to my advisory committee: Dr. Jim Greer, for being supportive of the research I wanted to do on top of working at the ULC. Dr. Julita Vassileva, for your many past publications that have inspired me, and your constructive criticism that motivated me to improve my writing. I also want to acknowledge you and other women I've met in CS. I think that women should be allowed to either succeed or fail at things (like programming!) without anyone assuming that their performance is related to being female. But when you're one of the only ones, this is really hard to avoid! Dr. Jay Wilson, for your feedback ranging from grammar to important questions about the human element when AI is applied to Education.

Thanks to others who have supported my studies: Dr. Mike Horsch, for the introduction to the wider field of artificial intelligence and insight into the reaches of AI with Prolog; Dr. Nate Osgood, for AnyLogic classes and introduction to system dynamics and agent-based modelling; Dr. Chris Dutchyn, for opportunities to improve presentation and research skills; Dr. Ray Spiteri, for the chance to look at one of my research questions under a new light (planning as an optimization problem); my colleague Dr. Carolyn Hoessler for the crash course on t-tests; Dr. Vania Dimitrova for helping me see how to connect real learning theories to my work, with Gord in Madrid! Thanks to my fellow students, lab mates, co-workers, friends, family members, and others, I can't even list all I have learned from and with you!

My research wouldn't have been possible without the support of my family. James, you are an awesome husband and dad to Sophia; your presence in my daily life brings me laughs and realness alike! Every week you gave me the solid back-to-back hours I needed for research. And Sophia, I admire you so much and am so lucky to be your mom. Because I wanted to be a good mom, I learned to separate "research time" and "family time" so I could dedicate full energy to both. Dad and Mom, thanks for encouraging my interest in CS and advancing my education. Thanks Kevin (and the tip about generative testing!), Laurier (I often think of our conversations about personality types!) and Cameron, who spent many hours helping Grandma (thanks Mom!) by entertaining Sophia when she came to visit, while I spent hours pouring over my simulations. I'm lucky that my mom is also a writer: thanks for reading a draft and finding places that needed clarification.

I also thank the staff of Oak Trees and Acorns at the Sherbrooke Community Centre. I would have given up on my dreams of research altogether if I hadn't been able to find a place for Sophia to go every day where I knew she would be safe, happy and loved. Most recently I appreciate the work of staff at St. Augustine School and the Boys and Girls Clubs of Saskatoon.

CONTENTS

Permission to Use	i
Abstract	ii
Acknowledgements	iii
Contents	iv
List of Tables	vi
List of Figures	vii
List of Abbreviations	ix
1 Introduction	1
1.1 DOELEs and the need for Simulation	2
1.2 The Ecological Approach architecture for DOELEs	7
1.3 Thesis Goals	9
2 Literature Review: Simulation in Advanced Learning Technology	11
2.1 Types of Simulation	11
2.2 Virtual Simulations are State of the Art	13
2.3 Constructive Simulations in Education and ALT	14
3 Exploring Effects of Peer Impact on Learning	18
3.1 Motivating problem	18
3.2 Simulation Model	20
3.3 Modeling Peer Impact	23
3.4 Experiment	25
3.5 Results	27
3.6 Conclusion	32
4 Instructional Planning for Dynamic Open-Ended Learning Environments	35
4.1 Instructional Planning for DOELEs	35
4.1.1 Approaches to Selecting what to do Next in ALT	36
4.1.2 Recommender systems	39
4.2 CFLS Planner	41
4.3 Simulation Model	43
4.4 Results	45
4.4.1 Baselines: Random Planner and Simple Prerequisite Planner	46
4.4.2 CFLS: what combinations of b and $f = s$ worked best?	47
4.4.3 Using the same b , did performance increase or decrease with changes to $f = s$?	51
4.4.4 Using the same $f = s$, did performance increase or decrease with changes to b ?	52
4.4.5 How did the CFLS planner compare to the Baselines?	53
4.5 Conclusion	53
5 Conclusion	57
5.1 Instructional Planning for DOELEs	57
5.2 Constructive Simulations of ALT	60
References	63

A	Peer Impact Simulations	73
B	CFLS Planner Simulations	80
C	Comparison of CFLS Planner Simulations	89
C.1	Changes in b	89
C.2	Changes in f	91
D	Prerequisite Graph used in Chapter 4	94

LIST OF TABLES

2.1	Classification system describing the relationships of people to simulations (Loper, 2015 [91]) .	12
3.1	Average P[learned] of six simulations in the fiftyfifty condition	30
3.2	Average P(learned) of six simulations in the mostlyrepelled condition	30
3.3	Average P(learned) of six simulations in the mostlyattracted condition	30
4.1	Baseline results (Random and SPP)	46
4.2	P-values for $b = 3$, $f = s = \{1...5\}$ (Excerpt from Appendix C: Table C.8)	50
4.3	P-values for $b = \{1...5\}$, $f = 3$ $s = 3$ (Excerpt from Appendix C: Table C.3)	52
C.1	P-values for $b = \{1...5\}$, $f = 1$ $s = 1$	89
C.2	P-values for $b = \{1...5\}$, $f = 2$ $s = 2$	90
C.3	P-values for $b = \{1...5\}$, $f = 3$ $s = 3$	90
C.4	P-values for $b = \{1...5\}$, $f = 4$ $s = 4$	90
C.5	P-values for $b = \{1...5\}$, $f = 5$ $s = 5$	91
C.6	P-values for $b = 1$, f and $s = \{1...5\}$	91
C.7	P-values for $b = 2$, f and $s = \{1...5\}$	92
C.8	P-values for $b = 3$, f and $s = \{1...5\}$	92
C.9	P-values for $b = 4$, f and $s = \{1...5\}$	92
C.10	P-values for $b = 5$, f and $s = \{1...5\}$	93

LIST OF FIGURES

3.2	Sample performance of an individual learner	27
3.4	Excerpt from Appendix A showing all 5 groups (Condition: fiftyfifty, hard mode)	29
4.1	Excerpt from Appendix B (Figs. B.1 and B.2) (b=1, f=s=1)	47
A.1	fiftyfifty, hard mode	74
A.2	fiftyfifty, easy mode	75
A.3	mostlyattracted, hard mode	76
A.4	mostlyattracted, easy mode	77
A.5	mostlyrepelled, hard mode	78
A.6	mostlyrepelled, easy mode	79
B.1	(b=1, f=1, s=1) score on Final Exam	80
B.2	(b=1, f=1, s=1) % LOs Mastered	80
B.3	(b=1, f=2, s=2) score on Final Exam	80
B.4	(b=1, f=2, s=2) % LOs Mastered	80
B.5	(b=1, f=3, s=3) score on Final Exam	81
B.6	(b=1, f=3, s=3) % LOs Mastered	81
B.7	(b=1, f=4, s=4) score on Final Exam	81
B.8	(b=1, f=4, s=4) % LOs Mastered	81
B.9	(b=1, f=5, s=5) score on Final Exam	81
B.10	(b=1, f=5, s=5) % LOs Mastered	81
B.11	(b=2, f=1, s=1) score on Final Exam	82
B.12	(b=2, f=1, s=1) % LOs Mastered	82
B.13	(b=2, f=2, s=2) score on Final Exam	82
B.14	(b=2, f=2, s=2) % LOs Mastered	82
B.15	(b=2, f=3, s=3) score on Final Exam	82
B.16	(b=2, f=3, s=3) % LOs Mastered	82
B.17	(b=2, f=4, s=4) score on Final Exam	83
B.18	(b=2, f=4, s=4) % LOs Mastered	83
B.19	(b=2, f=5, s=5) score on Final Exam	83
B.20	(b=2, f=5, s=5) % LOs Mastered	83
B.21	(b=3, f=1, s=1) score on Final Exam	83
B.22	(b=3, f=1, s=1) % LOs Mastered	83
B.23	(b=3, f=2, s=2) score on Final Exam	84
B.24	(b=3, f=2, s=2) % LOs Mastered	84
B.25	(b=3, f=3, s=3) score on Final Exam	84
B.26	(b=3, f=3, s=3) % LOs Mastered	84
B.27	(b=3, f=4, s=4) score on Final Exam	84
B.28	(b=3, f=4, s=4) % LOs Mastered	84
B.29	(b=3, f=5, s=5) score on Final Exam	85
B.30	(b=3, f=5, s=5) % LOs Mastered	85
B.31	(b=4, f=1, s=1) score on Final Exam	85
B.32	(b=4, f=1, s=1) % LOs Mastered	85
B.33	(b=4, f=2, s=2) score on Final Exam	85
B.34	(b=4, f=2, s=2) % LOs Mastered	85
B.35	(b=4, f=3, s=3) score on Final Exam	86
B.36	(b=4, f=3, s=3) % LOs Mastered	86
B.37	(b=4, f=4, s=4) score on Final Exam	86
B.38	(b=4, f=4, s=4) % LOs Mastered	86

B.39 (b=4, f=5, s=5) score on Final Exam	86
B.40 (b=4, f=5, s=5) % LOs Mastered	86
B.41 (b=5, f=1, s=1) score on Final Exam	87
B.42 (b=5, f=1, s=1) % LOs Mastered	87
B.43 (b=5, f=2, s=2) score on Final Exam	87
B.44 (b=5, f=2, s=2) % LOs Mastered	87
B.45 (b=5, f=3, s=3) score on Final Exam	87
B.46 (b=5, f=3, s=3) % LOs Mastered	87
B.47 (b=5, f=4, s=4) score on Final Exam	88
B.48 (b=5, f=4, s=4) % LOs Mastered	88
B.49 (b=5, f=5, s=5) score on Final Exam	88
B.50 (b=5, f=5, s=5) % LOs Mastered	88
D.1 Prerequisite Graph used in Chapter 4	94

LIST OF ABBREVIATIONS

ALT	Advanced Learning Technology
CFLS planner	Collaborative Filtering based on Learning Sequences
CMS	Course Management System
DOELE	Dynamic Open-Ended Learning Environment
EA	Ecological Approach
IP	Instructional Planning
ITS	Intelligent Tutoring System
LO	Learning Object
MOOC	Massive Online Open Course
SPP	Simple Prerequisite Planner

CHAPTER 1

INTRODUCTION

Advanced learning technology (ALT)¹ supports learners as they interact with the material, with instructors, and with each other. To support these interactions, ALT can prompt learners with new information or ask them a question using natural language. ALT can immerse learners in a 3D interactive environment, allowing them to explore, build, and experiment in virtual worlds. ALT can drill learners with fill-in-the-blanks exercises while offering timely hints. Most ALT uses a combination of these and other types of interaction. What makes ALT so impactful is it focusses on the individual learner: their goals, their characteristics, or their previous actions within the system.

This technology is made possible by ALT researchers who have created models of learners (called the *learner model*), models of the material being learned (called the *task domain model*), and models of different pedagogical approaches such as leading and guiding, or offering subtle feedback at just the right time. Together, these things are called the ALT system architecture. However, the complexity of ALT poses two main problems which I have studied in this thesis.

The first problem is that it is difficult make ALT available to anyone, learning anything online. ALT is usually limited to a certain task domain, to a certain approach to teaching, or to assumptions about what information should be tracked about learners. Whether the ALT be designed for math problems, science concepts, language learning or other, the ALT is usually quite focused because of the intensive modelling required. But for ALT to work more generally, researchers need to find a way to decouple the system architecture from the material that is being learned. A framework that addresses this problem is the ecological approach (EA) architecture (McCalla, 2004 [101]). This thesis is an exploration of how to build ALT that is designed for anyone, learning anything online, though use of the EA architecture. This is reflected in the thesis title as ‘System Design Issues’.

The second problem is represented in the title as ‘Pedagogical Issues’. The environment around the learners can have a big impact on their success, such as changes in the behaviour of other learners, the sheer number of other learners and their characteristics (e.g. novice or experienced), the difficulty level of the material, the amount of material present in the learning environment, or the amount of support available to

¹ALT is an umbrella term encompassing fields such as Artificial Intelligence in Education (AIED), Intelligent Tutoring Systems (ITS), Computer-Supported Collaborative Learning (CSCL), Learning Sciences, Educational Data Mining (EDM), Learning Analytics, etc.. The word ‘advanced’ means that these fields are pushing the boundary of the capabilities of mainstream learning technology.

learners. My ultimate goal is to design better ALT that is able to adapt to such changes. To do that, I need a way to study pedagogical issues in changing environments.

To explore these two problems, I’ve used simulation. Simulation helps engineers to design aircraft, and simulation helps meteorologists to study weather patterns. Simulation can also help ALT researchers to study system design and pedagogical issues. Just like an aircraft, ALT can be run in a simulation to test a change to the system architecture (system design issues). Just like weather patterns, simulations can trace different forces in an educational environment, and make predictions about the impact of certain interventions on learners (pedagogical issues).

The rest of this chapter is structured as follows. First, I describe a new term, *Dynamic Open-ended Learning Environment* (DOELE), which can be thought of as the open Web. Having framed my research area as designing ALT for DOELEs and studying pedagogical issues for DOELEs, I then argue for the importance of this work by showing that learners in DOELEs need the support of ALT. However, the complexity of ALT software architecture makes it difficult to get ALT to work in DOELEs. The EA architecture is ideal for this purpose, and I explain why (Section 1.2). Finally, I describe how the overall thesis is structured (Section 1.3).

1.1 DOELEs and the need for Simulation

We introduced the term *Dynamic Open-Ended Learning Environment* (DOELE) in a previous publication (Frost & McCalla, 2015 [52]). A DOELE is any collection of material that can be used for learning. A DOELE could be housed by an individual, a group, or institution. In a DOELE, there is no organizing structure that would need to be maintained as the contents change. The material can be linked from different sources and of varying quality. It is always changing because human knowledge evolves. Learners within a DOELE are also changing because they could enter or leave at any time, and all the while they are changing themselves as they are evolving their knowledge. The ultimate DOELE is the World Wide Web: anyone can upload anything at any time, and there is no central organization.

A DOELE is similar to, but not the same as, a “traditional” open-ended learning environment (OELE). The OELE is based on the constructivist approach to education (Hannifin, 1994 [63], Land, 2000 [84]) which, like DOELEs, has the spirit of giving learners choice in the problems they solve and how to solve them. However, OELEs are typically associated with a course that is fixed in its content, order and goals.

A traditional online course could be more or less DOELE-like depending on the flexibility that learners have in the order they view material, the content covered, and the ability to contribute new content. In a DOELE, everything is open-ended and continually evolving, including the learners and their goals, as well as the content, the order it will be encountered, and to what depth.

The Web in its current form works well for many learners in many situations, but there are times when more support is needed. Sometimes, the amount of material is overwhelming, the area is unfamiliar, and learners don’t know how to navigate. They might not be aware of a possible activity that would help them

overcome an impasse. They might not even know how to describe what they don't know and could be left floundering. Or, maybe learners need an extra push to challenge themselves to deepen their understanding. In these situations in DOELEs, learners could use the support of a knowledgeable guide.

This type of guidance could be provided by ALT. For example, a system could help to ease a learner's cognitive load by optimizing the order of activities (Corbalan et al., 2005 [30]). A type of ALT that manages personalized learning experiences is called an Intelligent Tutoring System (ITS). ITSs monitor the interactions with the learner both in the short term and the long term. In the short term, the ITS might ask the learner to answer a question or offer a hint. In the long term, the ITS guides what future content to cover at the topic level or even the curriculum level. The ability to monitor at the short and long term is known as the two loop description of tutoring systems, where the outer loop deals with the long-term and the inner loop deals with the short-term (VanLehn, 2006 [135]). Is critical that an ITS be personalized; that is, the system is designed to work individually with a learner to provide specific support according to the learner's current situation.

Mainstream online learners today find their guidance through Course Management Systems (CMSs) and Massive Online Open Courses (MOOCs). These systems lack the two loop structure that would provide personalized guidance. Instead, technology is relegated to the role of a medium to broadcast content, or as a communication tool for instructors and learners. Typically, material in CMSs and MOOCs is rigidly organized and coverage of the material is narrowed by necessity so that the instructional team can cope with the number of learners and the amount of material.

Unfortunately, the fruits of ALT research have not been incorporated widely in systems available to mainstream learners. In the literature, there is discussion of how to make ALT more widely available to mainstream learners. There are exceptions, such as the Cognitive TutorTM approach by Carnegie Learning² which have been used by thousands of students (Ritter et al., 2007 [124]). However, these are only exceptions and the ALT community has expressed the importance of sharing ALT with the wider field of Education (Underwood & Luckin, 2011 [134], Cumming & McDougall, 2000 [32]). When ALT is not widely seen or known, it can be difficult for ALT research findings to be communicated. Researchers have worked to close this gap (Brooks et al., 2006 [17], Brusilovsky & Vassileva, 2003 [20]). MOOCs offer many possibilities for bringing ALT to mainstream learners, and for providing researchers with data (Kay et al., 2013 [74]). Indeed, ALT researchers are working to bring MOOC developers up to speed (Siemens et al, 2014 [129]).

One of the ways that ALT provides personalization is in the area of control. *Learner control* is the learner's ability to choose the sequence, pace, flow or instructional style of the material being learned (Simsek, 2012 [130]). In mainstream systems, control is usually held only by the (human) instructor or sometimes only by the learner. In ALT, control can be held by the system itself or shared between the learner and the system.

ALT is carefully designed to effectively balance this control. Done poorly, learners won't be helped, or worse, might be inadvertently prevented from learning new things. For example, a "filter bubble" can be

²Cognitive Tutors, formerly known as Model Tracing tutors, are described in Section 4.1.1

created, where users become stuck with the familiar and are not exposed to new or differing knowledge. Filter bubbles happen when the system limits access to material based on the learner’s stated preferences. ALT can take steps to address filter bubbles such as by using visualizations to inform users of the filtering that is going on and giving them control over the filtering (Nagulendra & Vassileva, 2014 [112]). Limiting access to material is necessary to cope when there is too much material. But for learning to occur, the system also needs to be able to expose the learner to material that doesn’t match their stated preferences.

It may be tempting to place control only in the hands of learners themselves. But learner control isn’t always effective. In a meta-analysis of educational technology studies from 1996-2012 investigating the effects of learner control on learning outcomes, it was found that providing learner control had only a small effect and did not directly lead to increased learning outcomes (Karich et al., 2014 [73]). Learner control doesn’t seem to help when learners don’t have a goal in mind (Lawless and Brown, 1997 [85]). Even when learners are given recommendations by the system, the learners might ignore it. In a study in the context of OELEs where students were given a problem to solve and tools to help them solve the problem, the system’s advice had no significant influence on the learner’s usage of the tools (Clarebout & Elen, 2008 [29]). The authors suggested that giving advice might not work until after the student has gained enough background knowledge to be able to steer their own way.

When a teaching approach – such as providing high learner control – works very well for advanced students but the same approach has poor results with novices, this is called the *expertise reversal effect* (Kalyuga et al., 2003 [71]). In OELEs, learners sometimes require an outside push to move forward (Land, 2000 [84]). The same would apply to DOELEs. Providing this push, and determining when it is needed, is exactly what much ALT is designed to do.

Instructional Planning (IP) holds ideas that could be effective in DOELEs. IP is a core sub-discipline of ALT, especially in Artificial Intelligence in Education (AIED) and ITS (Peachey & McCalla, 1986 [118]). IP is concerned with managing learner vs. system control and selecting content, such as allowing experienced learners to skip over novice content.

IP grew out of the area of Artificial Intelligence called Planning, which is used for robotics. Robots are designed to work in cycle: pay attention to the environment, act to achieve the goal, observe the result, then make the next decision. IP is concerned with personalizing the learning experience, such as automatically adapting the sequence of learning activities to best suit the individual, whether it be a remedial action, or simply changing the order of activities. An instructional planner (or simply *planner*, for short) can tailor both the method of delivery and the content itself (Brecht, 1990 [16]). Another example of IP is the TOBIE system, which can generate maps of several possible plans, continually revising these as the learner progresses and changing its pedagogical approach as needed (system-led vs. learner-led) (Vassileva, 1995 [137]).

Thus, it seems probable that IP could be used to help to guide learners in DOELEs by working with the learner to select appropriate materials to cover within the DOELE and sharing control between the system and learner. The question is how to apply IP to work in DOELEs. Incorporating IP into DOELEs is a

challenge, because DOELEs don't have the constraints typical in most ALT systems.

An ALT system is normally constrained to work for specific courses that have been structured by course designers where course content doesn't often change. ALT requires a learner model, the task domain model, and the processes to implement the pedagogical approach of the system, such as the balance of learner control. A learner model provides the system with information such as the content the learner has already covered, their degree of success, their goals, and their preferences. An example of a learner model would be a profile that the learner completes before beginning the course. This structure allows the system to see how the learner's current understanding is related to each piece of content, and how the pieces of content relate to each other. Task domain models allow the system to navigate the content to be learned. To create these models, system designers or content experts create machine-readable versions of this content. This work is referred to as knowledge engineering. An example would be a prerequisite graph of facts about Geography or Mathematics.

Such knowledge structures are difficult to maintain because each time new learning material is added or removed, the structure needs to be updated by a system designer before that new material can be used by learners. Learner models are also difficult to keep up-to-date. Some learners might leave before completing their work in the system, while others may join long after other learners have already begun. The *open corpus problem* is that hyperspace (i.e. the Web) is too large and constantly changing for the necessary data structures to be in place to power ALT (Brusilovsky and Henze, 2007 [19]). A heavily knowledge-engineered approach doesn't scale well to DOELEs. In DOELEs, things don't stay stable long enough to engineer a knowledge structure that the ALT system can rely on.

The Semantic Web (The World Wide Web Consortium, [144]) is a major effort to provide infrastructure on the Web that can support advanced systems including ALT. A Semantic Web approach aims to re-use learning material, which takes time and effort to create. Educational material can be bundled into reusable learning objects (LOs) and annotated with metadata that can then be used by ALT to deliver the material to the learner in various ways. A LO is anything that can be used for learning and can range from very simple to having great depth. Examples of LOs include: a journal article, a diagram, a quiz question, a complete lesson module, or an interview with a person. In the IEEE Learning Object Metadata (LOM) Standard [1], LOs are labelled with information such as typical learning time, difficulty, typical intended age range of learners, interactivity level, as well as other aspects such as the author, file size, and copyright information. The Sharable Content Object Reference Model (SCORM) [86] builds upon the LOM standard to include rules for sequencing learning objects. This has evolved into the Tin Can API [87] which allows for educational systems to transfer data among each other.

In ALT research, it is common to adapt these standards or combine them with innovative approaches. For example, an instructional planner can construct plans that account for learning style and resource scheduling availability when the standard formats are expanded to include that information (Garrido & Onaindia, 2013 [54]). Learning objects can be marked up with prerequisite relationships using Protégé Axiom Language

(PAL) and used for a rule-based sequencing mechanism combined with a competency gap-filling approach (Shen & Shen, 2004 [127]). Metadata has been extended to include a new Instructional Role type to provide descriptors that govern the ordering of the material (Farrel et al., 2004 [47]). Web access logs have been used to recommend web pages to users using Path Clustering based on Competitive Agglomeration (PCCA), which expands on other machine learning approaches for web personalization and introduces the use of sequences for clustering users (Yu et al., 2006 [149]). This design assumes that people will access the most interesting choices first; therefore different orderings can be taken to indicate different user interests. Another research project, the SWARS (Sequential Web Access-based Recommender System) (Zhou et al., 2004 [154]), also uses web access logs to match a user’s sequence of web page visits to create a recommendation.

However, a problem remains with semantic web standards. It is very difficult to keep LOs annotated with enough metadata to be useful, and, unfortunately, most content is never annotated with this crucial metadata at all. The International Learning Object Metadata Survey found that many metadata fields were not even being used (Friesen, 2004 [51]), preventing ALT that relies upon this data from making effective use of the LOs. In response to this, approaches have been developed to automatically annotate LO metadata (Motelet & Baloian, 2006 [109] and Roy et al., 2008 [125]), and to automatically evaluate the quality of the metadata (Ochoa & Duval, 2006 [115]). The problem is even harder because the information on the Internet is changing all the time, and there is so much of it. New material (LOs) will always need to be added in response to changes in the course or the material, or to learner demand. Sometimes new material will be provided by the course developers, but other times, new material could be crowdsourced by the learners themselves and incorporated into plans (Hage and Aïmeur, 2008 [62]).

DOELEs create challenges not only for building ALT system architecture but also for studying pedagogical issues. An investigator may wish to study questions in learner space (different kinds of learners), evaluation space (different metrics for success), planning space (from rigid to reactive) and different kinds of courses such as using LOs at different levels on the Bloom taxonomy (Bloom, 1969 [11]) or courses focused on different breadths and depths of the prerequisite graph.

These kinds of questions are typically explored in ALT research by developing a system and then conducting a human subject study. However, a DOELE adds more variables and more uncertainty that can become unmanageable because human subject studies themselves often produce ambiguous results and are expensive to conduct. Human subject studies will always be needed, and are often the only way to uncover subtleties of human learning and experience. But to handle DOELEs, a different methodology is needed that can complement the prevalent approaches of today.

Thus, I’ve turned to simulation. I propose that simulation methodology ought to be used for developing and testing ALT for DOELEs, as well as studying pedagogical issues in DOELEs.

To make the argument that simulation helps ALT designers to develop teaching strategies, consider the aircraft designer who is testing a new plane in a wind tunnel. Just like the aircraft designer needs to adjust the velocity of the wind, the ALT designer needs to adjust things like the number and type of LOs and

learners in a DOELE. The simulation allows the designer to see how their system would work in different environments. The aircraft designer needs the simulation to report how well the plane is holding up under certain conditions, and the readings of the simulation need to reflect similar readings as if the plane were placed under the same conditions in the real world. Similarly, the ALT designer needs readings from the simulated learners using the ALT to correspond to how real learners would respond when using the same ALT system.

Any predictions that the simulation provides will need to be checked to see if it also comes true in the real world. Using simulation to develop ALT is still a new area. I can't claim that the simulations I've conducted in this thesis give results that are the same as if I had conducted these experiments in the real world. But I can claim that the simulations provide insight into the pedagogical and system design issues I've investigated, and these results provide a basis of comparison for future work.

Designing ALT with simulation means working in a cycle. The first step is to develop the simulation and run initial experiments whose aim is to narrow down many possible system designs to the most promising ones, as I do in this thesis. The result is a system design that holds up when used with simulated learners and LOs. The next part of the cycle is to check if the predictions of the simulation match the real world. For example, the simulated LOs can be replaced with real ones. Or, real learner data from a real course could be checked for evidence of having influenced each other like the peer effects I explore in Chapter 3. When the simulation's predictions match the real world, this provides confidence that the simulation model is a good one. Having a good simulation model allows designers to ask new questions by running the same model under different conditions.

Instructional designers of the future may one day use simulation as a standard tool for checking new course designs under different situations. They'll use real learner data to check their predictions, refine their course, run new simulations, and check their predictions against the real world again in a cycle to continually improve their designs.

To develop ALT using simulation, I've adopted a specific architecture, the ecological approach (EA) architecture, which is ideal for DOELE environments and will be discussed in the next section.

1.2 The Ecological Approach architecture for DOELEs

The ecological approach (EA) architecture (McCalla, 2004 [101]) is a framework upon which ALT can be built. Because the EA is compatible with DOELEs, ALT that is built on the EA will work in a DOELE. In this section, I describe the EA and how it has been used in the literature. I then describe three main challenges that ALT systems must overcome when using the EA, concluding that simulation can be used to address these challenges.

In the EA there is no overall course design. Instead, courses are collections of learning objects each of which captures usage data as learners interact with it. As learners interact with LOs, any information that

is known about the learner at the time of the interaction can be saved and associated with the LO. The EA assumes that each learner is represented by a learner model that contains static attributes (*characteristics*) as well as other data gathered as they interact with the LOs (*episodic*).

Over time the usage data accumulates and can be used for many purposes. For example, by running queries on the usage data, ALT can display information to learners or instructors about what learners are currently working on in the course. Another example of how usage data can be used is to save the usage data in a standards-compliant format so that it effectively automatically annotates the LOs (Miller et al., 2011 [106]). The ability to automatically annotate LOs would be useful if standards-compliant metadata were needed but there were too many LOs to apply the metadata manually.

Another use for EA usage data is that it can be used for instructional planning (Champaign, 2012 [26]). The work done by Champaign is the closest to mine in the literature. Using the EA, ALT can discover effective sequences of LOs based on the previous interactions of other learners (Champaign & Cohen, 2010 [27]). In this approach, a learner's knowledge is assessed before and after each interaction with a learning object. At the end of the interaction, these pre- and post- interaction scores are associated with the LO. This information is used to calculate how similar learners are to each other, and also to calculate the expected benefit of a possible sequence of LOs to provide to a learner. The authors validated this approach using simulation experiments.

An architecture to support DOELEs needs to be able to support the ALT system architecture while also permitting learners to come and go and the LOs to change. The EA captures information about the learner and how they have interacted with the LOs. This information is then used to support the ALT system architecture. No great knowledge engineering is needed for these to be incorporated because decisions will be made based mainly on the usage data. In addition, the EA allows learners to come and go because the only information required from learners is obtained whenever they interact with LOs and each other. The EA allows both the course material and the learning community to be open ended and dynamic.

The EA offers a promising infrastructure for developing scalable ALT. However, there are three challenges that need to be overcome when using the EA. First, many applications based upon the EA will suffer from the cold start problem. The cold start problem is that a system cannot be used until it has enough data, but the only way to obtain more data is for people to use the system. For example, suppose there were a real course with thousands of learners that was built upon the EA to capture learner interactions with the various LOs in the course. Assume that you wanted to build an instructional planner that would make use of all this interaction data. Even with thousands of learners, it could take several years for enough learners to build up enough interactions with each LO to provide useful data to inform the instructional planner. What takes the longest is the wait for learner who are in a minority - those who may have an unusual style of learning and are very few in number. Their usage data is needed to find the best way to help future learners like them, but it can be rare for such learners to appear, thus the system suffers from a lack of data.

Second, ALT that uses the EA will need lots of learners because the EA depends on lots of data. There's

a barrier for researchers without access to the personnel and funding required to recruit enough participants to generate enough usage data. In addition, ALT shouldn't be deployed to lots of learners unless it is fairly stable. In a course with thousands of learners, if there are problems while the ALT is under development, there is risk of causing confusion or inconvenience to a great many learners - "to fail at scale".

Third, it is not always possible to know how to set the conditions before the beginning of a study aimed to improve a planner. For example, there may be unanswered design questions such as the criteria to use for identifying an appropriate peer if one needs to be found, how long learners should be left to struggle on their own before the planner intervenes, and appropriate values for many other parameters that would be used by the system.

This thesis uses simulation to address these challenges.

1.3 Thesis Goals

The vision of this thesis is to enable ALT to be deployed in DOELEs which will guide learners to appropriate learning resources and support. To achieve this, I show how the EA architecture is ideal for representing DOELEs, so that when ALT that uses the EA, it can work in DOELEs. I demonstrate how simulation can be used to develop ALT for DOELEs and gain insights into pedagogical issues for DOELEs.

Chapter 2 is a literature review on how Education and ALT are already using simulation. Most simulations in ALT provide learners with new or unique learning experiences that would be impractical in the real world. A different type of simulation provides researchers with a way to develop and test their systems. This type of simulation is used more often in the social sciences, but has not been widely explored in ALT.

Chapter 3 explores a sample pedagogical issue with an EA-based simulation. Learners' own learning can be impacted by their reactions to seeing the performance of their peers in a DOELE. I describe the core elements that would be required to simulate a DOELE and point out what the simulation shows about peer impact. The core elements from this chapter are used again for the simulation experiment in Chapter 4.

Chapter 4 is focused on how to do instructional planning in a DOELE. For ALT to be able to run in a DOELE, the system architecture can't be hard coded to a centralized knowledge structure. Most ALT relies on knowledge structures that represent the material being learned, how it is connected to the learner model, and how to implement the pedagogical approach. A common knowledge structure is the prerequisite graph, which connects LOs together and is used by many ALT systems to determine the order LOs should be presented to the learner. However, a knowledge structure such as a prerequisite graph cannot work in a DOELE because the LOs would be constantly changing and thus the prerequisite graph would need to be updated each time. Knowledge engineering is labour intensive and too cumbersome for DOELEs. I've built a very simple EA-based instructional planner and use a simulation experiment to show how this planner is able to operate successfully without the need for a centralized prerequisite graph.

ALT researchers need faster, cheaper ways to explore issues with teaching and learning using technology.

The rare ALT that has achieved wide scale deployment usually has been based on decades of work to build up the necessary infrastructure. This thesis will show how to explore pedagogical and system design issues without the need to have this infrastructure in place for a real system, but instead using simulation. A simulation of the EA is representative of a DOELE environment and is thus a good environment in which to build ALT. When learners have the guidance of ALT, then human teachers don't need to plan ahead for every possible interaction their students may have online. ALT can also push and challenge learners in new and exciting ways.

CHAPTER 2

LITERATURE REVIEW: SIMULATION IN ADVANCED LEARNING TECHNOLOGY

Simulation is commonly used in Education and ALT, but not for testing and developing new learning technology. To understand the difference between common uses of simulation and the type of simulation I use in this thesis, this chapter provides an overview of the main types of simulation. Section 2.1 summarizes technical approaches and then elaborates on the different ways people can interact with, and are represented in, simulation. Section 2.2 covers the most common category of simulation used in Education and ALT research. Section 2.3 covers a different category that is most pertinent to this thesis but is most commonly used in other fields, like social sciences. The section ends with a discussion of model fidelity, that is, how detailed must a simulation be for it to be useful. Finally, this chapter shows some rare examples of others using simulation in the same way I do, that is, simulations that act as testbeds for developing ALT.

2.1 Types of Simulation

A simulation is a simplified version of the real thing working over time. A simulation can be considered a social simulation when it is used to represent person-to-person interactions, such as between learners. Social scientists can build an artificial society and use it to formalize their hypotheses and discover theories (Gilbert & Troitzsch, 2005 [57]).

The main technical approaches to simulation are time-driven, event-driven, system dynamics, and agent-based. A time-driven simulation executes certain actions at fixed intervals of time. An event-driven simulation reacts to certain events, regardless of the amount of time that might occur in between (Meyer, 2015 [105]). System dynamics simulations use stocks (counts of things) and flows (rates of change of things) to study feedback loops as parts interact with each other. Agent based simulations are composed of independently-acting agents that can send messages to each other and react accordingly.

The simulations underlying this thesis are agent-based. Research in this area studies how agents coordinate with each other to achieve the best performance. For example, agents can be designed to improvise outside of their normal role when necessary, thereby achieving desirable behaviour that didn't need to be explicitly designed for (Keogh & Sonenberg, 2014 [75]). Examples of agent-based simulations include: an urban traffic simulation where individual vehicles have agency (Mandiau et al., 2008 [93]), a system for personalized

information delivery to users in social network environments (Tandukar & Vassileva, 2012 [140]), and a trust-based service recommender system (Nusrat & Vassileva, 2013 [114]). Agent-based simulations have been used for developing an educational planner for a social assessment game (Laberge et al., 2014 [83]).

Loper (Loper, 2015 [91]) describes a classification system for the relationships of people to simulations. This classification system uses quadrants: whether people are real or simulated, and whether the system is real or simulated (Table 2.1). Using this classification system, I'll review some examples of simulation in education and training. A real person simulating a real system is called a *live simulation*. This could be a situation where professionals work together in a role playing scenario (Hood, 1997). A real person with a simulated system is called a *virtual simulation*. Examples of this type include a web-based simulation to help students understand concepts in economics (Nicholson and Westhoff, 2009 [113]), a simulation to help Engineering students to learn about digital logic and integrated circuits (Antao et al, 1992 [6]), and a simulated environment to train police officers about allocating police resources in a dynamic urban environment (Furtado & Vasconcelos 2007 [53]). Providing learners with an immersive virtual environment is one way to shift from transmitting theory in the classroom toward providing more inquiry-based, constructivist approaches to learning using simulation (Yaşar, 2004 [148]). The third quadrant, a simulated person in a real environment, does not have its own name in this classification system, but examples include a simulated patient used for medical education (Maran, 2003 [96] and Bradley, 2006 [13]), a generic simulated learner used for testing any real education system that has already been built (Virvou et al., 2003 [141]), and a simulated student being used by an instructional designer to develop and refine a real lesson they have designed (Mertz, 1997 [104]). The fourth quadrant, simulated people in a simulated environment, is called a *constructive simulation*, and is the most relevant category of simulation to this thesis. I discuss this category in detail in Section 2.3.

Table 2.1: Classification system describing the relationships of people to simulations (Loper, 2015 [91])

	Real people	Simulated people
Real system	live simulation	(no name)
Simulated system	virtual simulation	constructive simulation

Simulated learners have their own classification system. The three main types of simulated learners have been classified depending on who is interacting with or using the simulated learners(s): other learners,

instructors, or instructional developers (ALT researchers) (VanLehn et al., 1993 [136]). When a real learner interacts with simulated learners, they are often called simulated learning companions or *pedagogical agents*. They can play a variety of roles as they are embedded in an environment, as we will see in Section 2.2. A way that instructors can use simulated learners is to practise their tutoring strategies. A way that ALT researchers can use simulated learners is to test the design of their systems, as we will see in Section 2.3.

2.2 Virtual Simulations are State of the Art

Of the four categories that Loper described, it is virtual simulations that are most often used in ALT research. One meta study (Lee, 1999 [88]) distinguishes that (virtual) simulation can be used for two types of instruction: practice and presentation. Practice simulations require prior instructions outside of the simulation itself, whereas presentation simulations do not (Lee, 1999 [88]). Lee also defines pure simulations and hybrid simulations: pure simulations do not contain any embedded instruction or support whereas hybrid simulations do. Embedded instructional support is becoming the norm, with research focusing on the different effectiveness of that support. A meta study of simulation-based learning environments covered the effects of learner control and the effects of *when* assessment is conducted (after training, during, or both) (Gegenfurtner et al., 2014 [56]). Simulations can also be classified based on the type of educational value that the learner receives from the simulation (experience generation, conceptual understanding, skills development, and affective evaluation which is how the learner feels about the simulation experience) (Ranchhod et al., 2014 [121]). A creative use of simulation is to allow the learner to create the simulation itself. SimSketch is a tool for children to explore scientific concepts by creating them in the simulation using freehand drawings (Bollen and van Joolingen, 2013 [12]).

ALT research often pushes the boundaries of the roles typically played by humans or the system. In the late 1980s, the introduction of simulated learning companions (Chan, 1991 [28]) was a big step forward. Usually, the simulated person was a teacher figure. Simulated learning companions or peers offer different kinds of interactions to the human learner, such as competition and teamwork, with the possibility that the companion could be prone to making mistakes. Learners can strengthen their own understanding and get a sense of satisfaction by noticing and explaining these mistakes. Simulated learning companions can also help groups of students when faced with an unproductive situations, such as when one member is dominating the group or the group discussion goes off topic (Vizcaíno, 2005 [142]). Different pedagogical roles of the learning companions (expert, motivator, mentor) can lead to different changes in the learner (Baylor & Kim, 2005 [9]). In the Betty’s Brain System, learners are given the responsibility to help their companion to learn by providing it with information and observing its actions (Leelawong & Biswas, 2008 [89]), a “reciprocal learning” pedagogical paradigm.

Learners can interact in 3D environments with animated pedagogical agents such as Steve (Johnson et al., 2000 [69]) using visual communication that is not possible with 2D or text-based systems. Steve can nod

or point at objects, providing a richer interaction between the student and the system. AutoTutor (Graesser et al., 2005 [59]) is an agent that communicates with the learner using natural language while the learner is embedded in a 3D environment. Another example is the Tactical Language and Culture Training System (TLCS) (Johnson, 2010 [70]) which immerses learners in conversations with pedagogical agents to enhance cultural awareness. Systems such as Tactical French and Tactical Iraqi allow learners to practise face-to-face communication including nonverbal cues in a different cultural context.

Most of these simulations are aimed at skill acquisition. Questions have been raised in medical education about whether simulation can also provide opportunities for developing caring attitudes toward others (Diener & Hobbs, 2012 [35]).

Evaluation of virtual simulations is normally done with human subject studies, where one group of learners uses the ALT with a certain feature activated, while other learners use the ALT in a standard way or with enhanced features deactivated. Often there is also another group of learners not using the ALT at all. The outcomes, such as the learner grades in the course or score on pretests or post tests, are used as a measurement for the study. Sometimes the attitudes of the learners about the simulation are also used as measurements. Other times, evaluations are done using validation by human experts who accept or reject the system’s behaviour as a reasonable assumption of how a human teacher would act instead of the simulated teacher.

Human subject studies can be very expensive to conduct, and often produce noisy results. To help researchers cope with this expense, researchers can re-use high quality datasets from previous studies. Such datasets can be found online such as in DataShop (Koedinger et al., 2010 [81]). However, such datasets alone won’t cover all situations because the data can be deeply contextualized. Sometimes, researchers need to study a specific hypothesis that isn’t captured in the public datasets, so the researchers can’t use the datasets from previous studies. For example, many of the DataShop datasets follow the Cognitive Tutor paradigm, so those datasets would not be useful for a researcher looking for data about group interactions in OELEs. Sometimes, researchers need to be able to produce their own synthetic datasets that capture the context of the hypothesis being studied. Researchers can produce their own synthetic datasets using constructive simulations.

2.3 Constructive Simulations in Education and ALT

Simulation is intended to help investigators understand complex things. Just like learners who come to a better understanding by studying a simulation, ALT researchers can gain a prediction of how their system designs will work by simulating different scenarios.

Less common than virtual simulations are constructive simulations (simulated people in a simulated environment). Constructive simulations are most often used by educational administrators in planning organizational changes. Simulation is a way to keep track when many interacting parts are placed under different

conditions, producing different outcomes. This type of thinking, called systems thinking, has been proposed as a method to deal with the complexity of higher education (Mizikaci, 2006 [108]). Studies that use simulation to evaluate educational programs have been done as early as 1975 (Richards, 1975 [123]). More recently, in the Netherlands, a simulation was developed and validated to enable a faculty of economics in a higher professional education institute to shift from a rigid, centralized educational program to a flexible one that is more responsive to the needs of students (Shellekens et al., 2010 [126]). Simulation may also be used to assess multiple variables that were previously studied independently, variables such as the size of a school and its funding. A multi-agent simulation, for example, integrated these factors and helped to detect school network structures that were then used to evaluate policies and organization performance (Zhang et. al, 2014 [151]). Another example is for academic advising consultation booking (Orooji et. al, 2010 [116]). This study also developed tools to assist the designers of the simulated agents to evaluate and improve the system. Finally, simulations have also been used to better understand attrition bias, when participants leave a study prematurely (Dong & Lipsey, 2011 [36]).

Constructive simulations have potential to help with developing the teaching strategies embedded in educational software. A constructive simulation can simulate both the learners and the educational software itself as it is being used. This way, ALT researchers can try out the same approach on different cohorts of simulated learners and can try a variety of possible scenarios cheaply and frequently, even choosing outrageous conditions to test the edge cases of their software. This allows for the ALT researcher to gain a better idea of the range of possible behaviours of their system.

Constructive simulations can provide synthetic data in situations where real data is not readily available. A tool named Simulog generates simulated user logs to allow ALT researchers to see if their system is behaving as desired for different student profiles (Bravo & Ortigosa, 2006 [15]). Simulated learner data has also been used to overcome the cold start problem for a research paper recommender system (Tang & McCalla, 2004 [131]).

There are only a few examples showing how ALT researchers have begun to use constructive simulation for evaluating their systems. Simulation was used to detect a student’s learning style out of 16 possible alternatives because there were so many combinations that it was impractical to use human subjects (Dorça, 2015 [37]). Simulated learners were used in an adaptive testing system to evaluate the effectiveness of a mechanism (called an assessor) that gave a minimum number of questions to a learner to determine whether or not they had learned a topic (Abdullah & Cooley, 2002 [2]). Simulation was used to test a machine learning technique for selecting a teaching strategy on sequencing content (Iglesias et al., 2009 [67]). Constructive simulation was used to develop and refine the reward structure for trainees in a military training simulator (Alt, 2012 [4]). Educational recommender systems have also used simulation for evaluating whether the recommender can help learners with various objectives such as completing more activities in less time and using a greater variety of learning paths (Drachsler et al., 2008 [40]). Similarly, a learning path selection approach has been developed with simulation (Koper, 2005 [82]).

Constructive simulations are intended to help researchers to keep track of many moving parts. ALT researchers might need to test their system under variables such as few learners vs. many learners, advanced learners vs. novice learners, very active learners vs. lurkers, lots of learning material vs. sparse material, and strict system control vs. free exploration. Learner behaviour can be so complex that tools have been built to assist ALT researchers with shaping the system to give learners the best kind of experience. The InVis (Interaction Visualization) system creates a single visualization out of many individual playtest results of an educational puzzle game to give the designer insight into diverse learner experiences (Eagle et al., 2013 [44]). For scripted learning paths created with an interactive drama authoring tool, simulation has also been used to debug and refine diverse possible storylines (Medler & Magerko, 2006 [103]).

The idea of using constructive simulations to test software is not entirely new. In software engineering, there is a technique called generative testing which is essentially using constructive simulation for difficult-to-test software (Andrea, 2004 [5]).

Although there are a variety of examples of constructive simulation being used in Education and ALT, it is still very much a specialized niche of ALT research, not a mainstream activity. In contrast, the social sciences have long been using constructive simulations to understand complex systems. Simulation has become increasingly important in the social sciences with many frameworks becoming available (Lorig, 2015 [92]). These frameworks allow investigators to create models of the systems they wish to study, and run the models as a way to study the effects of various interventions that can be applied to the model. Simulation methodology has evolved to incorporate different ways to test for validity: with the real world (outcome validity), the expected conceptual result (process validity), and the absence of software bugs (internal validity) (Garson, 2009 [55]).

Even without the ability to fully model the richness of a human mind, advances have been made in human psychology and behaviour using simulation. One study implemented what is known about human perceptual mechanisms into a simulation model, then showed that this model could perform the same operations as the most powerful known algorithms from artificial intelligence (Cassimatis et al., 2009). Simulations have also helped uncover possible results of different actions in certain social situations, such as how to respond to a bullying problem (Pynadath & Marsella, 2005 [120]).

One possible reason simulation is not a fully adopted methodology in ALT is because of the issue of using high vs. low fidelity models. A high fidelity model captures great detail and likeness to the real world, whereas a low fidelity model does not. High fidelity models are time-consuming and expensive to build (if they're even possible) whereas low fidelity models can be quickly built. However, questions have been raised about whether low fidelity models can be of any value in studying the deep and complex nature of human learning. For example, one study found that a low fidelity method of generating simulated student data failed to adequately capture the characteristics of real data (Desmarais & Pelczer, 2010 [34]). When this simulated data was used for training a cognitive diagnosis model, the predictive power of the model was worse than when they used the simulated data that had been generated by a higher fidelity method.

If one's purpose were to use simulation for an accurate prediction of student behaviour, it has been noted that the fidelity of the simulation is important (du Boulay & Luckin, 2001 [42]). In medical education, it has been found that a high fidelity simulation assessment could be used to distinguish a learner as either novice or experienced (Girzadas et al., 2007 [58]). Another example of a high fidelity model is SimStudent which is able to model the actions of real learners so well that it is useful for predicting the performance of human learners (Matsuda et al, 2007 [98]).

But, are high fidelity models a requirement for creating deeply individualized learning environments? High fidelity models can't be developed for everything because there is so much uncertainty about the world. Indeed, in the social sciences it has been said that when studying such complex phenomena, it is important to keep the model as simple as possible (Axelrod, 1997 [7]).

There aren't many examples of low-fidelity constructive simulations being used to design and test ALT. But one example is a study where researchers using the KnowCat web-based collaborative document sharing system were able to perform systematic experiments that allowed them to see how different starting conditions (student attributes, difficulty level of the documents, etc.) led to different overall global results, giving insight into the global system behaviour (Barbero et al., 2007 [8]). Another example is the work by Champaign & Cohen, who developed a simulation model to validate their algorithm for selecting LOs based on the ecological approach (Champaign & Cohen, 2010 [27]). A final example is a project I was involved in that used very simple models but still gained valuable insights. In this study, we developed a constructive simulation based on the ecological approach and used data from two separate (unrelated) human subject studies. We were then able to use the simulation to make a meaningful prediction (Erickson et al., 2013 [46]).

Simulation will allow ALT to be strengthened from many iterations of design and feedback and avoid the risks of deploying ALT to real learners before it is thoroughly tested. Low fidelity models have led to useful discoveries in Education and shouldn't be dismissed. In the following two chapters, I use low fidelity constructive simulations to debug pedagogical and system design issues pertaining to large scale DOELEs.

CHAPTER 3

EXPLORING EFFECTS OF PEER IMPACT ON LEARNING

3.1 Motivating problem

By studying simulated learners in a simulated learning environment, ALT designers can observe how changes in the environment can impact the learners. Many things shape the environment: the number of other learners, the number of LOs and their characteristics, and the forces that guide the interactions between them, such as course requirements or behaviour of the ALT being used. Over time, and especially in a DOELE, the shape of the environment changes as the learners and LOs change. Through simulation, researchers can reach a better understanding of how these changes impact learners' experiences in the DOELE.

As a first test of using simulation to explore a pedagogical issue, I sought a problem in the Education literature that would allow me to explore the ecological approach (McCalla, 2004 [101]) in a simulated DOELE environment. The issue that I've explored is that of how learners impact each other's learning. At one time, it was commonly believed that learning occurred only between the learner, the material and the instructor, but it is now becoming better understood that learning is also influenced by other factors, including peers.

The study of peer impact is an active research area, with most literature indicating that peers have at least some impact on a learner's academic achievement. In the context of family and school factors, race and socioeconomic status, the results of one study suggested that students benefit from higher achieving schoolmates (Hanushek et al., 2006 [64]). Another study (in the area of reading and math) also found that learners benefitted from higher achieving peers, but less so for high achievers (Kiss, 2013 [80]). On the other hand, there is also research where no significant peer effects were found. A study based on student housing data found no significant difference in peer impact whether the peers were randomly assigned or socially proximate (Foster, 2006 [49]). Another study found no significant peer influences until they used models that controlled for gender, in which case they found that male learners strongly influenced each other's average academic rating, while female learners were found to be impacted by neither female nor male peers (Ficano, 2012 [48]).

Peer effects are often described in two categories: endogenous and exogenous effects. An *endogenous effect* is when the learner's outcome is impacted by their peers' outcome. For example, when fictitious information about past peer performance is presented, this can influence a learner's estimate of their own understanding (Zhao & Linderholm, 2011 [153]). An *exogenous effect* is when the learner's outcome is impacted by a

characteristic of their peer such as whether they are involved in sports or whether they live with their parents. The literature discusses the challenges of separating endogenous and exogenous effects to try and follow the influences that are impacting peer performance.

I've focused on endogenous effects because their complexity is well suited to be studied with simulation. Endogenous effects have a social multiplier because these impacts can propagate (Lin, 2010 [90]). When a learner's outcome impacts other learners, then their outcomes can change, which in turn affect the learner again, and so on in a feedback loop. Endogenous effects can go either way because peers can impact each other in both helpful and unhelpful ways. To track positive and negative effects and feedback loops such as this, I've used social simulation (Gilbert & Troitzsch, 2005 [57]).

Positive and negative peer effects are not fully understood, but one element is that of learner emotion. Interacting with peers or even just observing peers can lead to different emotions. Examples of learner emotions that might impact their performance include enjoyment, anger or boredom. Measurements for learner emotions in relation to their learning can be used such as the Achievement Emotions Questionnaire (Pekrun et al., 2011 [119]).

Lots of ALT encourages learners to interact with one another. With open learner modelling, learners can see system information about their own learning, and sometimes about other learners as well (Bull et al., 2007 [21]). This knowledge can impact their choices. For example, learners tend to participate in online learning activities only when they have trust in their peers or some degree of self confidence (Daniel et al., 2008 [33]). The Comtella system motivates users to increase their participation by using a visualization that enables users to compare and compete with each other (Vassileva & Sun, 2007 [138]). Learners can be explicitly guided to work together more effectively using Intelligent Collaborative Support Systems (ICSS) (Israel & Aiken, 2007 [68]). Learners can be shown open social student models where individual and group performance is visible, and even combined with system suggestions for topics to work on next (Hosseini et al., 2015 [66]).

Because learners do impact each other's learning, ALT researchers need to be careful about what information is provided to learners about their peers. The system should avoid forms of peer interaction that are known to be detrimental, and the system should encourage forms of peer interaction that are known to be beneficial for students.

However, it can be difficult to keep track of (let alone design ALT that accounts for) the desirable and undesirable social effects that occur. Course designers can control the material to be covered, but have much less control on the behaviour of the learners themselves. A different cohort of learners can lead to a very different experience, even with the same course material and the same instructor. How can researchers keep track of the impact on learning when it is affected by the cohort, such as other learners being novices or experienced, when the cohort keeps changing?

Simulation is used to study complex things, including the study of user impacts on each other's experience. Agent-based and system dynamics simulations of virtual communities have helped ALT researchers

to evaluate different approaches to managing user participation. In an agent-based simulation of virtual learning communities, investigators observed things like: once the community reaches a certain size, having non-contributing members isn't as harmful as one might think, and if managers want the community to continue to grow they need to introduce changes (Zhang & Tanniru, 2005 [150]). In another example, researchers simulated different variations of their system's incentive mechanism for encouraging user participation to find the best setting (Mao et al., 2007 [95]). Simulation is ideal for allowing ALT researchers to adjust certain variables and then observe the different possible results. These variables could be configuration settings of the ALT, or, these variables could be aspects of the environment such as the number of learners involved. For example, an activity plan could turn out very differently if a new group of learners joins an existing cohort, which could happen when online courses become more open and sharing occurs between online communities.

For my study¹ of peer impact with simulation, I've used the ecological approach (EA). The EA allows for the study of learners' impact on each other because each outcome of an individual's interaction with a learning object (LO) is captured and stored with the LO. In a given situation, one learner might receive a positive boost from a certain event (such as being inspired by their friend's success) while another learner might be negatively impacted under the same circumstances (such as feeling discouraged by or jealous of their friend's success). The EA allows for these outcomes to be tracked over time as part of the accumulation of usage data.

A simple simulation model could emulate many learners interacting with each other as they learn about a topic. This model would allow the study of patterns of positive or negative impacts as individual learners interact with each other. The following section describes the essential components of a constructive simulation to study the issue of peer impact, implemented in the EA.

3.2 Simulation Model

To simulate many learners interacting with each other, I used very simple representations of LOs and learners (a low fidelity model). The LOs and learners each have certain attributes which I denote in **bold**. Each LO has a **difficulty level**, represented with a number in the range (0,1) where higher values represent more difficult material. The simulated LOs are created at the start of the simulation, and their **difficulty level** does not change. Another attribute of a LO is its **prerequisites**. The simulation uses an acyclic directed prerequisite graph. If an edge is directed from LO₁ to LO₂, it means that learners should master LO₁ before going on to LO₂.

¹This research has been published in: Frost, S. and McCalla, G. (2013) Exploring through Simulation the Effects of Peer Impact on Learning. Proceedings of the 1st Workshop on Simulated Learners, International Conference on Artificial Intelligence in Education (AIED 2013), Memphis USA, July 2013, CEUR-WS.org, online CEUR-WS.org/Vol-1009/0403.pdf

Learners have a single attribute called **aptitude** which is a number between 1 and 10. The **aptitude** suggests how likely it is that they will do well. Learners might do better because they worked really hard, or because they have innate ability that makes things easier for them, or they have some other advantage that others do not. Whatever reason lies behind the likelihood of some learners performing better doesn't matter for this model. What matters is that the model captures that different learners bring different chances of success. It is assumed for this simulation that an individual learner's **aptitude** does not change as they learn, although, of course, their knowledge of the topic will.

The simulations that I've developed for this thesis can be described as time-driven because the simulations operate in steps. At each step, each learner interacts with one LO. It could be converted to an event-driven simulation where learners interact with LOs based on some kind of event system that is not tied to time steps, but this has been left for other work. My simulation is also an agent-based simulation because both the learners and LOs are implemented as agents.

The simulated learners flow through the simulation in parallel, being impacted by their peers as they go. At each step, each learner randomly consumes a learning object. Just imagine the learners gradually building up their successes or failures, with these results gradually accumulating around each learning object as per the EA. Now imagine a new simulated learner enters this environment. The learner, using ALT, might check to see the current class average, which the ALT obtained from the usage data currently attached to learning objects. The learner sees they are doing even better than the class average. Knowing this gives the simulated learner a 'boost' so that when they interact with another LO they receive a slightly higher score than they would have otherwise. But, suppose a great many other learners also receive a boost, thus causing a spike in the class average. As the simulated learner moves on to the next LO, the class average is recalculated and this time the learner observes that their performance is now lower than the class average, causing them to potentially become more depressed and thus possibly do a little worse. Both negative and positive feedback loops are possible in this social learning environment.

Depending on the state of the environment (the class average), and the learner's current score, certain situations can emerge. An unlucky learner's score might be driven lower and lower. Under different circumstances, the same learner might have caught a wave of boosts to help drive their score higher and higher with each LO interaction. The learner's score may or may not balance out depending on the nature of the material ahead and the behaviour of other learners. With such interactions influencing thousands of learners and LOs, the class average would fluctuate higher and lower, thus impacting the future performance of learners, which then impacts the class average again, and so on as the simulation continues over time. Because all of this is done in simulation, endogenous effects like this can be tracked and predicted, even in DOELEs.

Different simulations will generate different usage data, composed of hundreds or thousands of learner-to-LO interactions. This usage data provides valuable evidence for what the learners have done. One simulation might have learners visiting LOs randomly. Another simulation might have learners visiting LOs that have been specified by a selection algorithm such as an instructional planner. The two resulting sets of usage data

can then be examined for differences, such as whether learners did better in one case. This is the type of experiment I conduct in Chapter 4. In Chapter 3, learners visit LOs randomly with my focus being on the type of learners: How do learners perform when they have a different style of reaction to the performance of their peers?

The simulation needs a way to represent whether simulated learners are actually learning anything as they interact with the LOs. This is done using a numerical value called $P[\text{learned}]$. $P[\text{learned}]$ is a number between 0 and 1 that reflects the degree of success as result of an interaction between a learner and a LO, i.e. the “probability that the learner learned the LO”, or the “system’s belief that the learner knows the LO”. $P[\text{learned}]$ can be saved as part of the EA usage data that is associated with LOs after learners interact with them. In the real world, $P[\text{learned}]$ could be calculated using a quiz or by drawing an inference from log data of the learner’s behaviour while interacting with the LO. Each LO may have its own way of calculating $P[\text{learned}]$. For my simulation, the $P[\text{learned}]$ value is calculated using an *evaluation function* (Erickson et al., 2013 [46]).

The evaluation function can be tailored to generate $P[\text{learned}]$ in different ways depending on how the simulation is being used, or what is being studied. There are many things that could impact learning such as: the nature of the content being learned, individual learner characteristics, or social factors such as peer impact. I use the word *dimension* to refer to “one of the things that impacts $P[\text{learned}]$ ” that is part of the evaluation function. The evaluation function is a weighted sum, where each term represents one dimension (denoted in SMALL-CAPS) and a weight (denoted w_1, w_2, \dots). Any number of dimensions can be used in the evaluation function so long as each dimension is in the interval $[0,1]$ and the weights sum to 1.0.

I’ll go through an example because the evaluation function is an important part of this research. Equation 3.1 illustrates how an evaluation function calculates $P[\text{learned}]$ as a result of a learner interacting with a learning object. In this example, learning is considered to be a factor of three dimensions: the APTITUDE-OF-LEARNER, the DIFFICULTY-OF-LO, and whether the learner has already mastered prerequisite LOs, HASPREREQUISITES.

$$\begin{aligned}
 P[\text{learned}] &= (w_1)(\text{APTITUDE-OF-LEARNER}) + (w_2)(1\text{-DIFFICULTY-OF-LO}) + (w_3)(\text{HASPREREQUISITES}) \\
 &= (0.33)(0.1) + (0.33)(0.8) + (0.34)(1) \\
 &= 0.637
 \end{aligned}
 \tag{3.1}$$

The dimensions may seem similar to some attributes, but the difference is that dimensions must be in the interval $[0,1]$ and dimensions must give a higher value to reflect higher success. In this example, the learner has a low **aptitude**=1. To transform the attribute **aptitude** into the dimension APTITUDE-OF-LEARNER, divide the **aptitude** by 10. Transforming the **aptitude** in this way will fit the requirements of being in the correct interval and that higher values of APTITUDE-OF-LEARNER indicate a higher likelihood of success.

The learning object in this example is a fairly easy one with a **difficulty level** of 0.2. To transform the attribute **difficulty level** into the dimension DIFFICULTY-OF-LO, there is no need to divide because **difficulty level** is already defined to be in the interval [0,1]. However, a higher **difficulty level** does not indicate a likelihood of success; it's actually an inverse relationship. So, the inverse is taken $1 - 0.2 = 0.8$, which can be seen line 2 of Equation 3.1 where the dimension names have been substituted with the actual values.

In this example, the learner has already mastered the prerequisite LOs. The evaluation function knows about the prerequisite structure because each LO has an attribute (**prerequisites**) that points to the prerequisite LOs. The evaluation function checks the EA usage data surrounding the prerequisite LOs to see if the learner achieved a high enough value of P[learned] (for this simulation, a P[learned] of 0.6 or greater is considered to be mastered). Depending on what is found, a boolean value of either 0 (has not mastered) or 1 (has mastered) is substituted in for HASPREREQUISITES. The boolean value meets the requirements for a dimension because it's in the correct interval and the higher value (1 = has mastered) indicates a higher likelihood of success.

The weights in the evaluation function are fixed throughout the whole simulation. That weights are fixed is different from dimensions, whose values depend on the specific learner and LO involved at the time. In this example, the three dimensions are given approximately equal weight (0.33, 0.33 and 0.34) and meet the requirement that they sum to 1.

To conclude this example, P[learned] calculates to 0.637, which means that the learner is considered to have mastered the LO because P[learned] is higher than 0.6.

I said that the evaluation function can be tailored depending on what is being studied. An example of such tailoring would be to change the weights of the dimensions. To give higher weight to the aptitude of the learner, such as 60%, the new value could be $(0.6)(0.1) + (0.2)(0.8) + (0.2)(1.0)$, or 0.42. This value of P[learned] is lower than the original example (0.42 vs. 0.637), which is to be expected: Giving greater weight to this learner's (low) aptitude decreases the P[learned].

3.3 Modeling Peer Impact

This section describes how to model the effects that peers can have on one another's learning. In my simulation, learning is represented with P[learned], which is generated by the evaluation function. So, a new dimension is needed in the evaluation function to incorporate the influence of peers on P[learned]. I've called this new dimension PEER-IMPACT.

I created two styles of peer impact to capture that learners are impacted by their peers in different ways. Indeed, a learner's academic performance can be influenced by their social group membership (Wentzel & Caldwell, 1997 [146]). Because they are members of different social groups, two learners with the same **aptitude** interacting with the same LO might perform differently. For instance, if a learner is being positively

impacted by their peers, then the PEER-IMPACT dimension should cause P[learned] to be nudged higher. Like all dimensions, PEER-IMPACT produces a value in the range (0,1) to represent impact on P[learned]. The simulation experiment in this chapter uses the following evaluation function to compute P[learned] each time a learner visits a LO:

$$P[\textit{learned}] = .25(\textit{APT-OF-LEARNER}) + .25(\textit{DIFFIC-OF-LO}) + .25(\textit{HASPREREQS}) + .25(\textit{PEER-IMPACT}) \quad (3.2)$$

I chose to model peer impact on the individual learner's reaction to the current class average. The class average is the average P[learned] of all learners and can be obtained anytime from the EA usage data. I made the assumption that learners would have access to view the current class average at any time. Some learners might become encouraged when the class average is higher than their own, and perform even better than they would have otherwise. In the same situation, other learners might become discouraged and perform even worse. I created two types of peer impact inspired by system dynamics theory, which uses the concepts of a balancing feedback loop (going against the norm) and a reinforcing feedback loop (going in the same direction as the norm), where the norm is the current class average.

Each learner has a new attribute called **peer impact type** which could be one of two types, depending on how they react to the class average. Each learner is given one type at the start of the simulation and it remains fixed for each learner.

The first type is called *attracted-to-Peer-score* which is like a reinforcing feedback loop. This type of learner might be considered to be very empathetic: if peers are doing well (i.e. the class average is higher than the individual learner's), the learner will do better than they would have otherwise, but if peers appear to be performing poorly (the class average is lower than the individual learner's), the learner will do worse than they would have otherwise. This is a positive feedback loop, because as the learner performs better so does the class average thus further encouraging the learner to do better. There is a similar negative feedback loop if the class average is lower than their own.

The other, *repelled-from-Peer-score* might be considered a socially rebellious learner: if peers are doing well, this learner will do more poorly than they would have otherwise, but if peers are performing well, the learner will do even better than they would have otherwise. The definition of *repelled-from-Peer-score* matches the definition of a balancing feedback loop because when the class average is high, the learner's average goes in the other direction. When the class average is low, then the learner's score will be boosted higher than it would have otherwise.

Note the difference between **peer impact type** and PEER-IMPACT. The attribute **peer impact type** indicates whether a specific learner is of the type *attracted-to-Peer-score* or *repelled-from-Peer-score*. The dimension PEER-IMPACT gives a number between 0 and 1 to be used in the evaluation function. If PEER-IMPACT gives a value close to zero; the learner will do worse than they would have otherwise because the resulting P[learned] value will be lower. If PEER-IMPACT gives a value close to 1, the resulting P[learned] value will be higher.

Figure 3.1 shows how the value of PEER-IMPACT is derived. The values 0.2 and 0.8 were chosen as thresholds to allow clear effects of the two types of learner to emerge.

```

if currentLearner's peer-impact style is repelled-from-Peer-score
  if class average is HIGHER than mine
    set peerImpact == randomNumBetween(0.0,0.2)
  if class average is LOWER than mine
    set peerImpact == randomNumBetween(0.8,1.0)
if currentLearner's peer-impact style is attracted-to-Peer-score
  if class average is HIGHER than mine
    set peerImpact == randomNumBetween(0.8,1.0)
  if class average is LOWER than mine
    set peerImpact == randomNumBetween(0.0,0.2)

```

Figure 3.1: Function to generate PEER-IMPACT for a given learner at a given time in the simulation

In my simulation, all learners were considered to have the same degree of influence on each other. If one wanted the simulation to represent some peers having more impact than others (such as a learner being more heavily impacted by close friends than by distant acquaintances), something like a weighted social network could be used (Cela et al., 2014 [25]), but this has been left for other work.

3.4 Experiment

This experiment was implemented as an agent-based simulation using AnyLogic software (XJ Technologies [132]). At each step in the simulation, the 80 learners each visit one LO (selected at random). There are 100 LOs in total. The numbers of learners and LOs were chosen to reasonably represent a group of learners in a course, but to be small enough that the simulation wouldn't take very long to run.

There are six conditions in this experiment: three learner population profiles were each run in easy and hard modes. The learner population profiles are made up of different proportions of simulated learners having different styles of peer impact. In one profile, called 'mostlyrepelled', the learner population is comprised mostly of learners whose **peer impact type** is *repelled-from-Peer-score*. When the simulated learners are initialized, they have a high chance (80%) of being assigned the *repelled-from-Peer-score* personality and a low chance (20%) of being assigned the *attracted-to-Peer-score* personality. The other two population profiles are 'mostlyattracted', which is comprised mostly of *attracted-to-Peer-score* learners. The learner profile called 'fiftyfifty' is comprised of 50% *repelled-from-Peer-score* and 50% *attracted-to-Peer-score* learners (fifty-fifty).

Each of these population profiles was run in easy and hard modes. Easy mode means that the start of the simulation, the simulated LOs were created having a 60% chance of having the lowest or next to lowest possible **difficulty level**, and at the same time learners had a 70% chance of being assigned a high **aptitude** (8-10) at the start of the simulation. Hard mode means that LOs had 60% chance of having the highest or next to highest possible **difficulty level**, and at the same time learners had a 70% of being assigned a low **aptitude** (1-3) at the start of the simulation.

These six conditions were hand picked to be representative samples on a distribution of possible population mixes that should provide some insight about the effect of these two kinds of personality on the learning environment.² The model is deliberately stochastic; it produces slightly different results each time to give a better idea if the observed behaviour is due to just randomness or if the results are a reflection of the interacting relationships within the simulation. Each condition was run six times, giving thirty-six graphs in total. These are included in Appendix A.

To check whether any effects might occur from having the balance of the system shifted, at the halfway point in time 80 more simulated learners are introduced into the learner population. Something like this might happen in the real world if, for example two classes merged partway through a course, or if two study groups in an online course were mashed together, or due to the openness of many online courses (e.g. MOOCs) when new learners can join any time.

The main measurement taken in this simulation is the average P[learned] achieved by simulated learners on the LOs that they have visited. There are two ways to think of this average: the average of all LOs in the simulation (a global average), or only the average P[learned] of the LOs actually viewed by that learner (a local average). The local average is not a good way to compare the performance between learners because they may have viewed entirely different learning objects. For instance, one learner may have been given only easy LOs and another learner only given difficult LOs. For this analysis, I used the global average because the set of LOs is the same for all learners. Using the global average, all LOs are included, even those never seen by the learner. In the rest of this chapter, when I say ‘Average P[learned]’ or *performance measurement*, I am referring to the global average P[learned] value.

The performance measurement was recorded for each learner, then learners were gathered into five groups and an average performance measurement was recorded for each group. The first group, called the Class Average, is comprised of the entire population of simulated learners. The second group is comprised of those learners having the *attracted-to-Peer-score* personality, so I’ve called this the *attracted-to-Peer-score* group. The third group is the *repelled-from-Peer-score* group. The fourth and fifth groups are a breakdown of the *repelled-from-Peer-score* group by **aptitude**, i.e. high **aptitude** (simulated learners whose **aptitude**

²An additional condition was also designed, but ultimately not used in this thesis. The simulation has a flag called ‘suppression’ which alters the perceived class average of each learner. This was intended to simulate the possibility that learners might not want to share their progress with other learners, depending on their performance relative to the rest of the class (i.e. suppress their performance data from being displayed). The simulation was used to see whether the suppression effect altered the impacts of peers on learning. Simulations were run with this flag both on and off, resulting in 12 experimental conditions. However, setting this flag did not result in very different results. So, these were not included in the thesis to keep the number of graphs to a manageable size. All of the results in this thesis have the “suppression” flag set to false.

is between 8-10) and low **aptitude** (1-3). Note that the medium aptitude learners are not singled out as their own group and are only considered as part of the Class Average. The *attracted-to-Peer-score* learners were not broken down into their own groups by **aptitude**. These groups were left out to avoid cluttering the graphs with too many lines, but could be considered readily in immediate future work.

For this experiment, I wanted to see if there was a difference between the personality types and how they perform relative to each other under the different conditions.

This experiment looked for different performance measurements when the learner population is made up of different personality types (mostlyreinforcing vs. mostlyattracted vs. fiftyfifty) under different conditions (easy vs. hard mode). Simulation allowed for the comparison of these six conditions in a systematic way using the same number of learners and LOs, and the same number of interactions.

3.5 Results

Figure 3.2 shows the performance measurement for a typical individual learner (this one has **aptitude** = 2) for the duration of the simulation (time = 200). The vertical axis is in units of $P[\text{learned}]$.³ The horizontal axis represents time in simulation steps where the learner interacts with a single LO per step. The blue line (top) shows the average $P[\text{learned}]$ value of LOs seen so far (local average). This line is quite variable at the start because at $t=1$ only 1 LO has been seen so the blue line shows the $P[\text{learned}]$ for only 1 LO. As time goes on, the average $P[\text{learned}]$ is calculated over more and more LOs. The blue line stabilizes as time goes on with a growing basis of LOs seen by that learner.

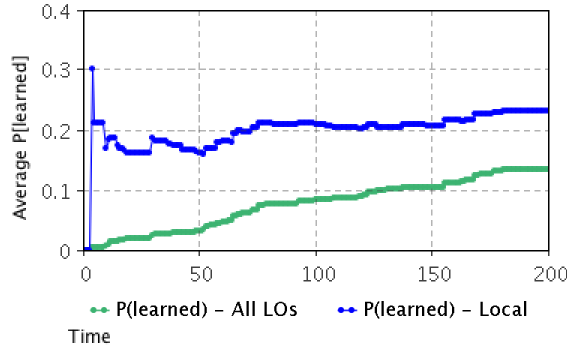


Figure 3.2: Sample performance of an individual learner with low aptitude (**aptitude** = 2)

³By definition, the value of $P[\text{learned}]$ is a number between 0 and 1. The reason the axis is cut off at 0.4 is that the average value never went higher than this. The performance is so low because learners were given LOs randomly. It could have been a very common occurrence for advanced LOs to be given to learners before they have mastered the prerequisite LOs, which would result in lower $P[\text{learned}]$ values being calculated by the evaluation function. Another reason for the low performance is due to the sheer number of LOs (100) created in the simulation and the time it would take for learners to visit them all. To spot check, the simulation was run again with only 30 LOs and with the same patterns observed, but with a steeper slope; the average $P[\text{learned}]$ reached around 0.5.

The green line (bottom) shows the global average $P[\text{learned}]$ that the learner has achieved so far at that point in time. Because this simulation does not allow for “negative” learning to occur (like forgetting or slips), and the base of LOs in the simulation was not changed in this experiment, the green line can only increase over time. A graph like this exists for each of the 80 simulated learners (but I’ve only shown this one example, Fig. 3.2) and gives an idea of what is happening for each learner. For some learners the green line is a very gentle slope, and for high aptitude learners the green line is a steeper slope because they are mastering more learning objects more quickly. If a learner were to interact with all LOs in the DOELE, the green and blue lines would meet.

The next graph, Figure 3.3, shows results averaged over groups of simulated learners. This graph uses the same horizontal and vertical axes, but the lines now only the global average $P[\text{learned}]$. The gold line (middle) is the average performance measurement for the entire learner population (the Class Average). The light blue line (top) represents the *repelled-from-Peer-score* group, and the red line (bottom) represents *attracted-to-Peer-score* group. The plot shows the average performance measurement of the individual learners in that group at that time step. The drop at time = 100 is when the new simulated learners were introduced into the learner population and is discussed later in this section.

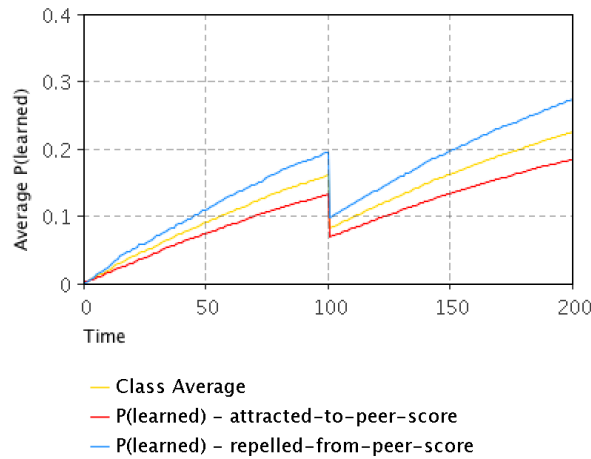


Figure 3.3: Sample simulation showing only 3 groups

At the start of the simulation, the class average is zero because no learners have yet interacted with any LOs. Note that the red and light blue lines represent subsets of the learners who are represented by the gold line. For the *repelled-from-Peer-score* group (light blue), their performance is higher than the class average (gold) because when the class average is zero, regardless of how well or poorly they do on their first LO, their score will almost certainly be higher than zero, so they will receive a boost and score higher than they would have otherwise. Consequently, the *repelled-from-Peer-score* learners increase their scores faster than normal because of the boost from the peer impact. But, even the learners in the *attracted-to-Peer-score* group increase their scores over time, just not as quickly. This is because in the same situation - when they see the class average is zero and their own grade is higher - these learners will have their grades held back

to be more similar to the grades of their peers who aren't doing as well. Over time, the gap only widens because the repelled-from-Peer-score learners continue to be encouraged and they excel, driving the class average even higher. Whenever a learner in the *attracted-to-Peer-score* group does manage to increase their score above the class average, they will immediately be pulled back down. This same behaviour keeps the *attracted-to-Peer-score* learners from falling too far behind, because when their scores are lower than the class average, they receive a boost to help them keep up.

The next figure (Fig. 3.4) shows all 5 groups of simulated learners to be examined in this experiment. The light blue line has now been broken down into the two remaining subgroups: high **aptitude** *repelled-from-Peer-score* learners (thick purple line) and low **aptitude** *repelled-from-Peer-score* learners (thick dark blue line). The experimental results consist of 36 graphs like this one, included in Appendix A.

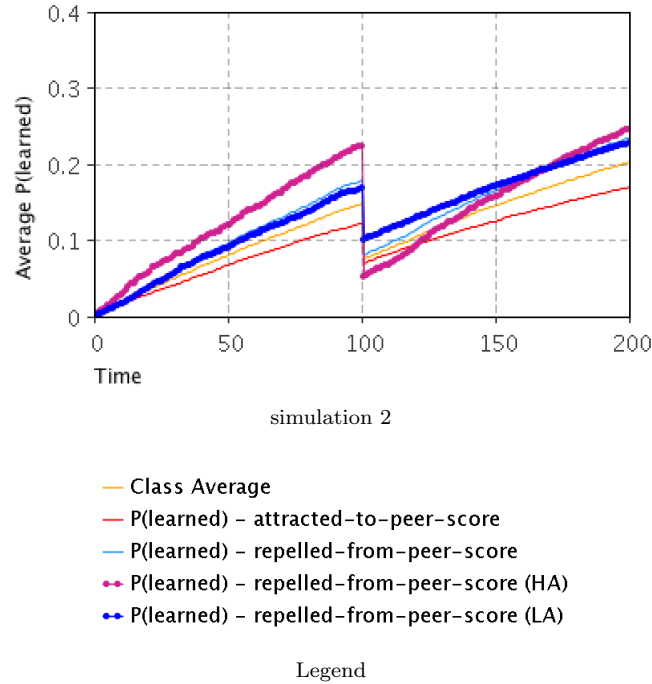


Figure 3.4: Excerpt from Appendix A showing all 5 groups (Condition: fiftyfifty, hard mode)

Each page in Appendix A shows one of the six conditions, each of which was run six times to show the possible variation given the randomness of the simulation. For each group of learners, the average of the six simulations is used to compare groups of learners between conditions and are summarized in Tables 3.1, 3.2 and 3.3.

One might expect that learners performed better (i.e. that the average $P[\text{learned}]$ was higher) in easy mode than in hard mode. Looking at the average $P[\text{learned}]$ for all learners (Class Average, gold line), in all cases the class average was indeed higher in easy mode than in hard mode. The difference in class average between easy and hard modes was .055 in the mostlyrepelled condition, .061 in the fiftyfifty condition, and .067 in the mostlyattracted condition.

Table 3.1: Average P[learned] of six simulations in the fiftyfifty condition

Group of simulated learners	easy mode	hard mode
Class Average (entire population)	.263	.202
attracted-to-peer-score Learners	.230	.167
repelled-from-peer-score Learners	.307	.248
repelled-from-peer-score Learners - High aptitude	.336	.255
repelled-from-peer-score Learners - Low aptitude	.238	.247

To avoid clutter in the graphs, attracted-to-peer-score learners are not broken down by aptitude.

Table 3.2: Average P[learned] of six simulations in the mostlyrepelled condition

Group of simulated learners	easy mode	hard mode
Class Average (entire population)	.283	.228
attracted-to-peer-score Learners	.232	.167
repelled-from-peer-score Learners	.308	.242
repelled-from-peer-score Learners - High aptitude	.342	.252
repelled-from-peer-score Learners - Low aptitude	.240	.243

Table 3.3: Average P[learned] of six simulations in the mostlyattracted condition

Group of simulated learners	easy mode	hard mode
Class Average (entire population)	.247	.180
attracted-to-peer-score Learners	.232	.160
repelled-from-peer-score Learners	.313*	.247*
repelled-from-peer-score Learners - High aptitude	.342*	.253*
repelled-from-peer-score Learners - Low aptitude	.242*	.250*

* Asterisk indicates these groups are quite small in number. Even though the values are the average of six simulations, these values may be less reliable because in the mostlyattracted condition, the repelled-from-peer-score learners are about 20% of the population, with the High and Low aptitude groups being further subdivisions of this minority.

Why would there be the greatest difference between easy and hard modes in the mostlyattracted condition? In easy mode, the class average would be higher, and so the *attracted-to-Peer-score* learners would receive a boost if their own score was not as high as the class average yet, thus driving the class average even higher. In hard mode, the class average would be lower, and so any *attracted-to-Peer-score* learners who were at the top of the class would have their performance driven lower, thus bringing the overall class average down even more.

Looking at the *repelled-from-Peer-score* learners, by definition these learners do well when they are doing better than the class average (and do worse when they are doing worse than the class average). This pattern showed itself in the extreme with low **aptitude** learners of the *repelled-from-Peer-score*. These learners actually did better on average in hard mode than they did in easy mode. This can be seen in all three tables (3.1, 3.2 and 3.3), the average P[learned] value is higher in hard mode. Without simulation, it is unlikely this effect would have been observed, yet it is a logical consequence of the many factors involved.

The Class Average was visibly higher in the mostlyrepelled condition than the mostlyattracted condition, regardless of whether in easy or hard mode. This seems to suggest that in the learning environment being simulated, it was more advantageous to be a *repelled-from-Peer-score* learner because the class average was higher when most learners were of the type *repelled-from-Peer-score*. However, when comparing the performance of the *repelled-from-Peer-score* group between the mostlyrepelled and mostlyattracted conditions, there was not much difference ($0.313 - 0.308 = 0.005$ in easy mode and $0.247 - 0.242 = 0.005$ in hard mode, i.e. the value in Table 3.3 subtract the value in Table 3.2). When comparing the performance of the *attracted-to-peer-score* group, there was no difference between mostlyrepelled and mostlyattracted in easy mode. In hard mode there was more of a difference (higher by .007 in the mostlyattracted condition), but this doesn't account for the large difference in Class Average. In retrospect, it is difficult to compare groups of learners based on averages of averages as this may have resulted in a loss of information. Different statistical measurements such as a t-test may have helped to more thoroughly explore trends for these different subgroups.

At the halfway point in time, there is a visible drop in the average P[learned] for all groups in all conditions. Such a drop did not occur for individuals (as in Fig. 3.2). This drop is only visible when looking at the performance measurements averaged over groups of learners. This drop was expected because the introduction of new learners, who start with a grade of zero, certainly would bring down the average grade.

What is interesting about the disruption at the halfway point is that the change in class average grade caused different effects on the learner's performance, depending on their style of **peer impact type**. In all of the hard mode conditions, a phase shift occurred. A phase shift is when two groups of learners have settled into a pattern of relative behaviour - that is, the *repelled-from-Peer-score* learners with high **aptitude** are performing better than *repelled-from-Peer-score* learners with low **aptitude**, that is, the purple line is above the dark blue line - and then this pattern is disrupted. At the halfway point in the simulation, suddenly the purple line has fallen below the dark blue line. The change in environment led to a group of low aptitude learners to be achieving higher than a group of high aptitude learners. The influx created a situation where

there are now learners with high averages intermingled with learners with zero averages. A different grade distribution creates a different environment than the starting condition where everyone started at zero. For low aptitude learners, suddenly their average was higher than the class average, and having a higher average gave a boost to those of the *repelled-from-Peer-score* style. For high aptitude learners whose average was suddenly lower than the class average, having a lower average caused them to achieve lower P[learned] values than they would have otherwise. (The definition of *repelled-from-Peer-score* states that when a learner is doing worse than the class average, then they will do even worse.) However, the high aptitude learners who were doing worse than the class average eventually caught up despite the way the current environment caused them a disadvantage. Looking at the graphs in Appendix A, the purple line catches up and often surpasses the blue line, thus returning the system to its previous equilibrium state prior to the disruption.

The phase shift did not occur in easy mode. This is likely because learners had already achieved high enough performance measurements that the introduction of new learners and drop in class average didn't change any learner's perception of their own performance enough for it to shift the balance. Because there are so many factors involved, even with these very simple models of learners and LOs, simulation is invaluable for studying these sorts of changes. Different environmental conditions cause the model to exhibit different behaviour, and different parts of the evaluation function can dominate at different times.

3.6 Conclusion

This study provided a better understanding of how learning is impacted over time by the influences of other learners and their interactions with learning objects. When thousands of learner-to-LO interactions occur, it would have been difficult to predict whether the current environment would lead to a wave of success or whether it would make things harder for learners to succeed. The simulation managed the details of each learner's success being impacted by the difficulty of the LO, the aptitude of the learner, whether the learner has mastered prerequisite LOs, and by the learner's perception of how other learners are doing compared to themselves. Simulation also gave the ability to systematically try different conditions: the makeup of other learners (mostly attracted or mostly repelled) and the nature of the material being learned (easy vs. hard).

The essential components of a simulation to study such issues are: simulated learners with an **aptitude** and style of **peer impact type** (*attracted-to-Peer-score* or *repelled-from-Peer-score*), simulated LOs with a **difficulty level** and **prerequisites** connecting them, an *evaluation function* to calculate P[learned] which is saved as usage data in the EA, and finally, a behaviour. The behaviour used in this chapter was random; learners visited a random LO once per time step.

Simulation was crucial for uncovering behaviours that might have otherwise gone unseen. I will highlight three findings that could inform future work in the real world. Findings such as these could be used as a hypothesis for future exploration in a more sophisticated simulation experiment or a real world experiment.

First, when comparing the Class Average between easy and hard modes, as expected, the Class Average

was higher in easy than in hard mode, regardless of the makeup of the learner population (mostlyattracted, mostlyrepelled or fiftyfifty). The difference in learner performance between easy and hard mode was taken for each population. The greatest difference occurred for the mostlyattracted condition, when most learners were of the *attracted-to-Peer-score* type. That there was a greater difference during the mostlyattracted condition could correspond to a hypothesis for the real world. An instructor may find a greater difference in Class Average between two modules (e.g. an easy and a hard one) in learners who have the *attracted-to-Peer-score* type than in those learners of the *repelled-from-Peer-score* type. In turn, knowing there would be a greater difference for *attracted-to-Peer-score* learners could lead to a recommendation on how to target support to individual learners. For example, more success stories from peers should be highlighted for *attracted-to-Peer-score* learners because seeing their peers succeeding would perform better.

Second, the *repelled-from-Peer-score* learners were found to perform their best during the mostlyattracted condition. This suggests that these type of learners may perform best when surrounded by peers who are different from them. In the real world, knowledge that *repelled-from-Peer-score* learners perform best in this case could inform future work on how to best cluster learners for group activities. *Repelled-from-Peer-score* learners should be placed in groups that are made up of a majority of *attracted-to-Peer-score* learners.

Third, in the phase shift, there were some high **aptitude** learners (of the type, *repelled-from-Peer-score*) who were performing at a lower level on average than certain low **aptitude** learners (of the type, *repelled-from-Peer-score*). A group of high **aptitude** learners performing worse than a group of low **aptitude** learners occurred when new learners entered the environment and disrupted the Class Average, which then changed the way learners performed on subsequent LOs. In the real world, knowledge that this situation could occur could reveal a possible reason why students who were expected to do well (because of their high aptitude) did not (because of the combination of their style of peer impact and the current environment).

A broader impression from this study is that when studying peer impact in a DOELE, it is important to recognize how the current DOELE environment impacts learner success. Whether the DOELE is made up of learners who are all starting at the same time or whether learners are arriving at different times, these things play a role in predicting a learner's success. None of the three results I described were expected, but all of them make sense when studying the usage data and taking all the conditions into account. In the real world, unexpected things happen all the time, and simulation is one way to try many possibilities and discover and explain phenomena that have been heretofore unknown.

In this analysis, sometimes it was difficult to tell if the Class Average was being influenced by one of the subpopulations. I used averages of averages which may have obfuscated some detail, such as outliers, which might have better explained the difference between subpopulations. To better track such things in Chapter 4, I switched to using t-tests to compare simulated learner populations.

Future work can also look at more experimental conditions and more detailed measurements. For example, the easy vs. hard modes could be expanded into four combinations of high/low **aptitude** learners and high/low **difficulty level** of the learning objects. In this experiment, I only broke down the *repelled-from-*

Peer-score learners by **aptitude** and could readily study *attracted-to-Peer-score* learners in the immediate future.

My motivation to better understand pedagogical issues is that I want to enable the design of better ALT. ALT needs to be able to predict how the specifics of the current DOELE will impact individual learners. If an ALT system can do this, it could intervene when it detects the current Class Average will push a learner into a situation where they are likely to perform worse than they would have otherwise. The system could more prominently highlight information about peers that is likely to motivate the learner rather than discourage them. Ultimately, it should be possible to shape the environment toward a configuration that is most beneficial for everyone. The ability to productively shape the environment is especially important when designing ALT for wide scale use, where many learners will be affected by the design decisions that have been made.

CHAPTER 4

INSTRUCTIONAL PLANNING FOR DYNAMIC OPEN-ENDED LEARNING ENVIRONMENTS

4.1 Instructional Planning for DOELEs

Most ALT developed over the last 40 years has required some degree of knowledge engineering. In a DOELE, the engineering exercise can be confounded at any time because the knowledge being engineered could be removed or changed at any time. For ALT to work in DOELEs, researchers need to decouple their system designs from the LOs within the DOELE. To explore the problem of how to select the next thing for learners to work on in a DOELE, I've developed an ALT system, a simple instructional planner, the CFLS planner (Collaborative Filtering based on Learning Sequences). This planner does not rely on the need to explicitly knowledge engineer the prerequisite relationships between the learning objects nor to annotate the learning objects with metadata about their content or usage. Rather, it draws on the inferences made from the interactions of learners with content. The ability for the planner to draw inferences is enabled by the ecological approach architecture, allowing the CFLS planner to work in DOELEs. In addition, the CFLS planner is among the first planners to use collaborative filtering to recommend sequences rather than just recommending individual LOs.

In developing the CFLS planner, I wanted to explore balance of control issues in instructional planning. I wanted to see the impact on learning using various parameters of the planner: when the planner is configured to plan far ahead vs. only a step or two, when the planner is configured to re-plan frequently vs. infrequently, and when the planner is very strict vs. liberal when selecting neighbourhoods of peers during the collaborative filtering phase. To prove that such a planner works would be difficult in the real world, but the clarity and simplicity of simulation has allowed me to clearly demonstrate the effectiveness of this approach to planning, as well as for tuning these parameters to see how the CFLS planner would behave under different conditions.

I deliberately took the stance that the DOELE could have any structure. I considered LOs to be independent of one another, but made no assumptions about their grain size. One LO could encompass a whole module, or it might encompass only a narrow concept. My goal is to build learning tools to support the learners within such a wide open structure.

The chapter is organized as follows. To gain an appreciation of how knowledge engineering has been used

over time in ALT to solve the pedagogical problem of what to do next, I'll summarize previous instructional planning approaches in Section 4.1.1. The pedagogical problem of what to do next has also been investigated using recommender systems research, especially when applied in learning technology, which I summarize in Section 4.1.2. In Section 4.2, I describe how the CFLS planner works, and Section 4.3 explains how this planner can be placed within a simulation where the learners and the LOs are simulated as they interact with the CFLS planner. This section also explains the conditions of the simulation experiment and describes measurements that are taken. Section 4.4 describes the baseline results and compares them with various settings that the CFLS planner was run under. Section 4.5 concludes this chapter and highlights areas for future work.

4.1.1 Approaches to Selecting what to do Next in ALT

In the early 1990s, the HyperCard system (Shin, 1994 [128]) explored the issue of a learning system's architecture, such as whether the task domain is organized in hierarchical or network format, and how this format ultimately impacts the experience available to the learner. The question about data formats led to a broader study about learner control. The system's architecture impacts what the system itself is capable of doing, and in turn impacts the learner's experience.

Researchers have long known about the danger of creating a rigid experience where the learner takes a passive role. To keep learners involved, early computer-assisted instruction (CAI) developed pre-programmed maps of possible sequences of content, essentially a hard-coded outer loop. This enabled systems to insert interactions between screens displaying content such as posing a question for the learner to answer. The content to be shown next would be determined by the learner's response, so that the learner could be asked to review remedial material or told to move on to more advanced concepts. The map data structure included the questions to ask the student as well as branching criteria for moving between content depending on the learner's responses. An example of this kind of system is PLATO (Programmed Logic for Automatic Teaching Operations), a generalized tutoring system that could support any subject matter. Lessons were written in a variety of subjects such as Biology, English, Mathematics and Music, using the programming language TUTOR (also known as PLATO Author Language). PLATO was found to have achieved high acceptance largely because instructors had a lot of control over its use (Murphy & Appel, 1977 [110]).

The pre-programmed maps in early systems required that all branching possibilities be thought of by ALT researchers beforehand. For such early systems to work in a different task domain, a new pre-programmed map would need to be developed. Researchers exploring ALT sought to find a way for systems themselves to be able to compute the next step. Systems with this ability would be more generalizable and could work with different subject areas without the need to reprogram for each subject. Generalizable systems would be able to react to unexpected actions from learners that had not been explicitly programmed for. To make this leap, researchers looked to the use of artificial intelligence techniques (McCalla, 1992 [100]).

Model Tracing tutors appeared in the 1980s and are now called Cognitive Tutors. These systems incor-

porate a cognitive model of learner problem solving in the tutoring domain. Model tracing systems such as the Genetics Cognitive Tutor (Corbett et. al, 2010 [31]) don't need to have pre-programmed each possible response that could be displayed to a learner. By monitoring the learner's actions, the system can immediately act, such as offering a hint, based on this underlying model and the learner's input. The system has an internal model of correct ways to solve a problem, and also has programmed in possible common errors, so whenever a learner appears to be going off track the system can react immediately. This allows for a far greater number of possibilities than earlier systems such as PLATO because the system actually has understanding of the material the learner is trying to learn and can therefore offer very specific hints. However, learners are not allowed to stray from known paths to solutions. In addition, it is difficult and expensive to create the cognitive model that is required for Model Tracing tutors to operate (Blessing et al., 2009 [10]).

In most educational areas it is unrealistic to build a complete library of common errors, since there are multitudes of possible incorrect solutions and too many to represent explicitly. The Constraint-Based Modeling (CBM) approach was first implemented in an SQL Tutor (Mitrovic & Ohlsson, 1999 [107]). With CBM, it is only necessary to represent constraints on correct solutions. The system knows that the learner has made a mistake when the learner's solutions violate these constraints. This type of representation gives more flexibility to learners because it doesn't matter in which order they enter their solution - as they work on their solution, the system can provide feedback even when there is no single correct solution that could be represented. In addition, because the CBM approach doesn't need to model every possible learner action, it can be inherently more scalable to other subject areas because there is not quite so much knowledge engineering that needs to be ported, i.e. only the representation of correct solutions, rather than detailed processes for solving problems and pre-determined possible errors.

Another environment where learners have lots of freedom is the Betty's Brain system. The system is based on a concept map that the learner builds into Betty's head as the learner is trying to teach Betty the content that the learner themselves is supposed to learn (i.e. learning by teaching Betty). This system is able to react based on what the learner has put into the concept map. Rather than explicitly building in the possible reactions, the Betty's Brain environment collects logs of everything the learner does. Using these logs, sequence mining techniques were used to analyze learner behaviour patterns to better understand and respond to the learner, such as with scaffolding techniques (Kinnebrew et al., 2013 [79, 78, 77]). The term *scaffolding* is used when learners have great freedom but the system is still able to provide support.

While just-in-time hints or suggestions can be valuable, there is also need for guided support that reaches several steps into the future. Long ago, researchers suggested the use of planning technology from artificial intelligence to build systems that could do longer term decision-making and adaptivity (Peachey & McCalla, 1986 [118]), creating the field of Instructional Planning (IP). These systems often have a detailed learner model to keep track of the learning activities that have already been completed as well as an established plan for achieving a particular goal. As the learner interacts with the system, the planner provides guidance

for the next activity to be completed by the learner, whether this be moving toward a goal or exploring a previous topic in a new way to address a misconception. Instructional planning can support many different pedagogical approaches (Wasson, 1996 [145]). For example, some approaches, such as inquiry-based learning and constructivism, require that learners have great freedom within the environment to ask questions at any time or take the time to experiment on their own without being interrupted or rushed into something else at the system’s direction. Instructional planning can be used to implement a particular teaching strategy, such as taking control and giving a demonstration or, staying back and simply giving a hint. Examples of systems that use planning to intelligently blend different teaching strategies to appropriate situations are PEPE and TOBIE (Wasson & Vassileva, 1996 [139]).

The more a system knows about the learner’s current understanding and about whether learners are even following along, the better a system is able to support them. Decision theoretic planning has been used to deal with this uncertainty such as unexpected learner behaviour or other situations (Matsuda & VanLehn, 2000 [99]). A decision theoretic instructional planner needs to have: a list of possible states (i.e. possible situations with imminent expected behaviour from the learner), the expected utility value associated with each state, a list of things that could happen that would change the state (such as possible system actions and possible learner actions), and the probability that these things would lead to other states. A policy table is a mapping from states to actions, i.e. given a state, the policy table tells the planner which action to take next. Often several actions are available and the planner chooses the action that leads to the highest likely utility. An advantage of this planner is that it can handle sudden and unexpected situations while still being able to guide the learner through a plan that is likely to help them achieve their goals effectively and productively using a refined set of actions.

Semantic databases allow several different data sources for subject matter to be merged together to give learners a coherent experience over a greater amount of information. To support learner exploration, a technique called *nudging* encourages learners to follow paths that the system believes would be most beneficial using two aspects “knowledge utility”: the learner’s familiarity (or unfamiliarity) with the topic, as well as the knowledge density (i.e. number of nodes connected to a certain topic) (Al-Tawil et al., 2013 [3]).

User generated content such as comments or posts are known as *digital traces*. Digital traces are interesting because they are dynamic and cannot be knowledge engineered ahead of time by ALT researchers. Yet, the information created by users in digital traces can be a valuable source for many pedagogical purposes, as in the ecological approach. The patterns implicit in digital traces (or usage data) must be discovered after the content has already been created. A process called semantic augmentation has been applied to digital traces to create a sort of map that can be used to organize and learn from the vast amounts of digital traces (Karanasios et al., 2013 [72]). The augmented semantics can then be used by the ALT system.

The word *stigmergy* is used outside of ALT when agents communicate with each other by leaving messages for each other in the environment. The term was coined by a scientist who studied termites that communicated with each other by leaving chemical messages to each other in the environment (Grassé, 1959 [60]). This idea

has been used in robotics in modern times, such as by robots leaving messages within a floor (embedded with RFID chips). The messages within the floor are later read by other robots who collaborate with each other to achieve goals without the use of a centralized plan or method of communication (Khaliq et al., 2014 [76]). Stigmergy allows for advanced coordinated behaviour to emerge in dynamic and uncertain environments (Ricci et al., 2007 [122]). Such approaches have similarities to the EA, where messages are left as usage data and kept with LOs.

In this section, I attempted to show how the underlying architecture and techniques of ALT has contributed to the system’s ability to adapt to the learner. All of the systems described are deployed in relatively well-defined domains (subject areas) because their complex data structures and processes can’t be easily extended into a DOELE. In the 2000s, a different technology came into great popularity. Recommender systems are much better suited for many domains and are able to bring personalized results to the user, and indeed have reached widespread mainstream adoption.

4.1.2 Recommender systems

A recommender system has three parts: information about the items to be recommended, information about the user (such as preferences, ratings, keywords to describe their interest), and an algorithm that combines information from the previous parts to create recommendations (Burke, 2002 [23]). Burke also explains how types of recommender systems can be categorized according to their emphasis or use of these parts. For example, a content-based recommender would rely on having attributes or tags for the items to be recommended, while a collaborative filtering recommender would rely on ratings from users. Several methods for combining content-based and collaborative filtering approaches to create hybrid recommenders are also presented by Burke.

Recommender systems can deal with diverse content because the underlying data needed by the system is created incrementally as the system is used. As users browse items or curate their own profile, they create the ratings data and content tag information that is needed by the system. By relying on ratings, a system doesn’t have to rely on knowledge engineering. These approaches have been compared to each other with simulation experiments, with some results suggesting that a ratings-based system can be a good alternative to a knowledge engineering approach for some uses (Nadolski, 2009 [111]).

The idea of leveraging usage data dates back to early case-based approaches (McCalla et al., 1980 [102]). One recommender system using the case-based approach is The Wasabi Personal Shopper (Burke, 1999 [22]) where users can also refine and change their query while viewing interim results in an ongoing dialogue with the system. The case-based approach has also been used to create a self-improving instructional planner that continually changes based on how it has been used by the learners (Elorriaga, 2000 [45]).

Early recommender systems were most often evaluated using accuracy metrics, which measure how closely a recommender system’s prediction of a user’s preferences match the user’s actual rating of each item. In recent years the field of recommender systems has shifted toward adopting other measures such as item

coverage, confidence metrics to support user decision-making, the learning rate, how long an algorithm needs to gather data before producing good recommendations, or novelty/serendipity for measuring whether a recommendation is new/unexpected (Herlocker et al., 2004 [65]). Recommending good sequences of items is not yet well explored (Herlocker et al., 2004 [65]) but finding good sequences is important for Education. The order that a learner interacts with the items can result in dramatically different learning outcomes. For example, a learner’s success in interacting with an advanced learning object can be greatly impacted if the user has mastered a prerequisite learning object before the advanced learning object rather than afterward.

In Education, a recommendation can take numerous forms. Learners can be given recommendations of the learning object to work on next. For example, a recommender system was developed that recommended LOs that the learner was likely to rate highly, with the system giving consideration to the competencies currently being targetted for that learner (Cazella, 2010 [24]). Learners can be given recommendations of other learners who could help them with a specific problem. Recommending another learner was done in the I-Help system (Greer et al., 2001 [61]). Learners can be given recommendations of teammates to join their learning-oriented group (Brauer & Schmidt, 2012 [14]). Or, instead of recommending something to an individual learner, a recommendation can be tailored for a group of learners (Dwivedi and Bharadwaj, 2015 [43]).

Recommender systems have also assisted learners with navigation through course materials. For example, traditional navigation offered in a MoodleTM[39] course was augmented with a hybrid recommender system so that in addition to browsing the learning activities in standard order, learners were also given a suggestion of a specific learning activity to do next (Drachsler et al., 2009 [41]). In this system, each user had a profile that was used to create neighbourhoods of similar users and included information about their motivation, study time, as well as their interests. These interests were also associated with items that could be recommended. To generate a recommendation, whenever possible the system would suggest learning activities preferred by others in the same neighbourhood. If not enough information was available to provide the peer-based recommendation, then the match was based on linking the user’s interest with the topics of the learning activities. Some knowledge engineering was done to enable the system to recover whenever there was a lack of data to create a recommendation. The results of this study showed that users receiving recommendations had increased variety in the order that learning activities were completed, and, it took less time to complete the same number of learning activities.

Many more educational recommender systems have been developed with around 40 separate recommender systems summarized (Manouselis et al., 2011 [94]). Standards-based approaches have also been used for creating neighbourhoods by exploring how the number of LOs from a target learner’s recent browsing history should be used for creating the neighbourhood (Zhang, 2013 [152]). New ways of collecting data specifically for educational purposes have been explored, going beyond using tags or attributes and instead tracking when learning objects (LOs) are accessed by a learner and then inferring that learning objects that are accessed around the same time by the user are related to one another (Orthmann et al., 2011 [117]).

Recommender systems are widely applicable but aren't able to support learners as closely as Model Tracing tutors, Constraint-Based tutors, or other ALT seen in the previous section. Recommender systems generally do not have an inner loop for working closely with the learner on a particular concept.

4.2 CFLS Planner

In this section I describe a new kind of instructional planner (called the CFLS planner - Collaborative Filtering based on Learning Sequences) that recommends sequences of LOs to learners. The CFLS planner is related to what recommender systems do in that it uses as little knowledge engineering as possible. Traditional instructional planners use knowledge structures that would be too cumbersome to maintain in DOELEs. But the CFLS planner is suitable for DOELEs because it does not rely on a centralized knowledge structure; rather it uses the ecological approach architecture. In the EA it is not necessary to have explicit models of learners, knowledge engineering of the domain (i.e. subject area), nor rules that implement a pedagogical approach. The EA captures what is known about learners and the outcome of each LO interaction; then this data is used by the CFLS planner to make inferences about learners and what they have learned.

The CFLS planner creates plans using a collaborative filtering approach. A plan is a sequence of LOs to be recommended to a learner, who wouldn't necessarily need to follow it, depending on the preferences of the system designer. To find the sequence of LOs, the CFLS planner looks for sequences that worked for similar learners in the past. The sequences that worked for other learners are found by drawing on the usage data created in the EA. Successful paths for particular types of learners, regardless of whether they follow standard prerequisites, is the only criterion of success. Like biological evolution, new learners or new learning objects will find their niche - some paths will work for some learners but not for others, and the niche is discovered automatically through usage. I address the cold start problem toward the end of Section 4.3.

The CFLS planner works as follows. For a given target learner the CFLS planner looks backward at the b most recent learning objects consumed. The variable, b , represents how far back to match the target learner's history. For example, if we want to take the current learner's three most recent learning objects and find peers who have viewed those same three learning objects together, then $b = 3$. Note that the neighbourhood could include learners who exited the DOELE long ago, but their usage data remains to benefit future learners. When the usage data is captured, a timestamp is recorded so the order of the LOs visited can be re-created, even if months or years have passed since the interactions occurred.

To create the neighbourhood of peers who have visited the same b most recent LOs, I allowed different permutations to be considered a match. This simplification was needed to create a critical mass of similar learners that are taken as the target learner's neighbourhood. For example, if $b = 3$, and the target learner recently viewed LO₁ (pass, i.e. $P[\text{learned}] \geq 0.6$), then LO₂ (fail, i.e. $P[\text{learned}] < 0.6$), then LO₃ (pass), then a peer who had the following sequence in their history would be considered a match: LO₂ (fail), LO₃ (pass), LO₁ (pass). Note that the pass/fail outcome must match for the specific LO. The following sequence

would *not* be considered a match: LO₄ (pass), LO₂ (fail), LO₃ (pass), because not all three learning objects match.

The higher the value of b , the more similar the learners in the resulting neighbourhood, but also the more unlikely it is to find enough or any matching learners to add to this neighbourhood. A very low value of b , such as $b = 1$, would create a neighbourhood for the target learner where all peers interacted with one LO in common and achieved a similar result. A neighbourhood created with a low b could be a very large group with wildly different learners who just happen to have had the same outcome on the same most recent LO. If no peers can be found, for this experiment the CFLS planner will select a random learner to be used as a peer instead.

Next, the planner looks forward at the f next LOs traversed by each neighbour and picks the highest value path, where value is defined as the average P[learned] achieved on those f LOs ahead. The highest value path is then recommended to the learner, who must follow it for at least s (for “sticky”) LOs before replanning occurs. Of course, s is always less than f . This approach provides a starting point for studying the balance of control between the learner and the system in terms of how frequently to re-plan when the system has control. If s is quite high, then the system is in control for a long period of time without re-planning. If s is quite low, then the system has control for only a short time before re-planning occurs. In this experiment, I used $f = s$ and tried several different combinations of b and f to find which leads to the best results. “Best results” can be defined in many ways, but I’ve focused on two measurements that were taken for each learner at the end of each simulation: the *percentage of LOs mastered*, and the *score on the Final Exam*. The score on the Final Exam is taken as the average P[learned] on a select few advanced LOs (the leaves of a prerequisite graph connecting the LOs that is unavailable to the planner, shown in Appendix D) interpreted as the ultimate target concept, which in the real world might well be final exams. The goal is to see if learners can still succeed on the “final exam” when following the sequences of LOs recommended by the CFLS planner, which must recommend those sequences without knowledge of the underlying prerequisite structure.

How well does the CFLS planner work? This will be explored in a simulation experiment which will determine if the learning outcomes are better for the CFLS planner than they are for a baseline planner, or no planning at all (e.g. random LOs). In this experiment, using a fixed a population of simulated learners and a fixed set of learning objects, I compare the results between different methods of selecting the next learning object. The three methods are selecting LOs randomly (Random Planner), and two different instructional planning algorithms: a Simple Prerequisite Planner (SPP) and the Collaborative Filtering of Learning Sequences (CFLS) approach.¹ There are three groups of simulated learners, one group using the SPP, one using the CFLS planner, and one using Random LOs. Each learner visits the same number of

¹This research has previously been published in: Frost, S. and McCalla, G. (2015) An Approach to Developing Instructional Planners for Dynamic Open-Ended Learning Environments. Proceedings of the 2nd Workshop on Simulated Learners, International Conference on Artificial Intelligence in Education (AIED 2015), Madrid, Spain, CEUR-WS.org, online CEUR-WS.org/Vol-1432/sl_pap1.pdf.

learning objects, regardless of the group they’re in. To compare the groups, two measurements (discussed above) were taken on the P[learned] values of the relevant LOs (the result of a learner’s interaction with each LO) that gradually build up in the ecological approach during the simulation.

4.3 Simulation Model

The simulation used in this experiment is based on the model in Chapter 3. As in Chapter 3, simulated learners have an attribute called **aptitude**. This time, I did not use **peer impact type** because I am tuning the simulation to the particular questions I am exploring, so the learners in this simulation are different than those in Chapter 3. Learning objects again have an attribute called **difficulty level**. An evaluation function is used to determine P[learned], the probability that a learner has learned the LO. The simulation executes the evaluation function each time a learner interacts with a LO (once per time step) and the resulting P[learned] value is associated with the LO. The evaluation function takes into account the aptitude of the learner, the difficulty level of the LO, and whether the learner has mastered the prerequisite LOs HASPREREQUISITES, if applicable.

The evaluation function also has a new dimension, SEENBEFORE. Sometimes, it is more likely that a learner will master the LO if they have seen it before. Perhaps the concept was not understood the first time but it made sense the next time around. The dimension SEENBEFORE returns 0 if the learner has never seen the LO before, returns 0.1 if it has been seen once before, 0.2 for twice before, and so on up to 1.0 for ten times before. I added SEENBEFORE because I wanted another dimension, beyond HASPREREQUISITES, that influences P[learned] depending on the LOs previously consumed by the learner. The order of LOs consumed impacts the learning, and prerequisites aren’t the only thing that influences whether a sequence will be successful.

The dimensions and weights of the evaluation function used in this experiment are as follows:

$$\begin{aligned}
 P[\textit{learned}] = & \\
 & (0.2)(\textit{APTITUDE-OF-LEARNER}) + (0.1)(\textit{DIFFICULTY-OF-LO}) + \\
 & (0.5)(\textit{HASPREREQUISITES}) + (0.2)(\textit{SEENBEFORE})
 \end{aligned}
 \tag{4.1}$$

I’ve chosen the particular weights assigned above to be sure the LO ordering would have impact, with the weight of HASPREREQUISITES set quite high at 0.5. The simulated learners are more likely to fail if the LO selection algorithm feeds them LOs that deviate from prerequisite order, but it is still possible for learners to master them. To master LOs that are given to learners outside of prerequisite order, learners would need to have a high **aptitude**, and the **difficulty level** of the LO needs to be low enough, and it would help if the learner has seen the LO before. I considered a LO to be mastered when $P[\textit{learned}] \geq 0.6$. I didn’t pick a higher threshold like 0.8 because it would make it too difficult for low aptitude learners to progress. The lower threshold also allows for non-prerequisite orderings to be successful.

I emphasize that the CFLS planner has no knowledge about the underlying prerequisite structure of the learning objects. The separation of the CFLS planner from the underlying prerequisite structure is critical for CFLS planning to work in DOELEs. There is still a prerequisite graph included in the simulation because the evaluation function uses it when computing values for $P[\text{learned}]$ because of the `HASPREREQUISITES` dimension. But the CFLS planner knows nothing about the prerequisites; it only sees the $P[\text{learned}]$ outcomes as captured in the EA usage data. When simulated learners are replaced with real learners in an actual DOELE, the evaluation function would disappear and the value for $P[\text{learned}]$ would come from a real world alternative, such as quizzes or an inference about the learner after their interaction with the LO.

This experiment has three conditions: the CFLS planner, the SPP, and a Random planner. In each of these conditions, simulated learners are grouped into 3 groups: low, medium and high aptitude learners. The SPP and the Random planner serve as baselines. The Random planner simply recommends learning objects randomly. The SPP assumes that a centralized data structure, a prerequisite graph, is available for the instructional planner to draw upon for choosing the next sequence of learning objects, and delivers LOs to learners in prerequisite order. The CFLS planner makes no such assumption and instead must make the decision of choosing which learning object to select next based on the EA usage data that is created as a result of previous learners interacting with learning objects. This experiment asks whether the decentralized approach to instructional planning, the CFLS planner, can work as well or better than the SPP.

As with all collaborative filtering approaches, the CFLS planner relies on having usage data from other learners. Thus, the simulated learning environment needs to be in operation for some time before the CFLS planner is introduced, and then it can be “launched” using interaction data from previous learners. By default, the simulation starts with an empty history - no simulated learners have yet viewed any LOs. To initialize the environment, the SPP was used to select learning objects for learners for some time for the EA usage data to accumulate. The EA usage data from the SPP was saved as a synthetic dataset and used to initialize the case base before the CFLS planner was brought in. A new population of simulated learners (with identical characteristics as the learners who interacted with the SPP) was then brought in to use the CFLS planner. The interaction data from the learners who interacted with the SPP was then compared with the interaction data of learners who used the CFLS planner to compare the two approaches to planning. This simulation experiment was aimed at seeing if, with appropriate choices of b and f , the CFLS planner could work as well or better than the SPP. Because the evaluation function is the same for the SPP and the CFLS planner, any differences in the overall results between the SPP and CFLS approach will be a result of learners being given learning objects in a different order.

There is still a cold start problem even after the simulation has been initialized with interaction data from the SPP. The cold start problem remains because the simulated learners who are to follow the CFLS planner have not yet viewed any LOs themselves, so there is no history to match the b LOs to create the plan. In this situation, the CFLS planner matches the learner with another random learner (from the interaction data from the SPP), and recommends whatever initial path that the other learner took when they first arrived in

the course. Another solution could be to start new learners using the SPP, but I didn't do this because I wanted to keep the CFLS planner entirely separated from the underlying prerequisite structure, relying only on the EA usage data.

The CFLS planner relies on the data captured from previous learners. In this experiment, the CFLS planner used data from learners who had followed the SPP. The CFLS planner could have instead launched itself with usage data from learners who had visited LOs randomly, although there would be little inherent information in the usage data so the simulation would likely have to run much longer before meaningful results could be expected. The advantage is that the CFLS planner can discover new paths that don't need to be restricted to a prerequisite ordering but can still be effective for the learner.

The most computationally expensive part of the CFLS planner is finding the learners in the neighbourhood, which is at worst linear in the number of learners and linear in the amount of LO interaction history created by each learner. The way the CFLS planner creates a neighbourhood is different from a typical recommender algorithm, where each learner is either matched with each other learner or each LO. The CFLS planner first narrows down the neighbourhoods when each learner's LO interaction history is searched to check for a match with the last b LOs. The forward searching of the next f LOs is then executed using only the small resulting neighbourhood.

To set up the aptitude groups, simulated learners whose aptitude was between 1-3 were placed in the low aptitude group, 4-7 in the medium aptitude group, and 8-10 in the high aptitude group. Two performance measurements were taken for each simulated learner: their percentage of LOs mastered, and their score on the Final Exam. These measurements were averaged for learners in each aptitude group, giving an average % LOs Mastered and average score on Final Exam for each aptitude group, for each condition. The same values for the simulated learner attributes were used for all simulation runs (i.e. CFLS, Random and SPP).

The questions I've sought to answer with this experiment are: 1) What choices of b , f , and s lead to the best learner performance, for different aptitude groups? 2) Are there any relationships between these variables and learner performance? (for example, does learning increase as b , f , or s increases?) 3) Did the CFLS planner lead to better learner performance than the SPP, or the Random planner?

4.4 Results

In this experiment, there were 65 simulated learners, grouped by aptitude. In the low aptitude group, there were 21 learners, in the medium aptitude group 26 learners, and the high aptitude group had 18 learners.² There were 40 LOs, each with a difficulty level and possible prerequisite relationships with other LOs.

²Originally, there were an equal number of learners in the low and high aptitude groups. However, I found a typo that resulted in invalid data for several learners, so the usage data generated by those learners was discarded.

4.4.1 Baselines: Random Planner and Simple Prerequisite Planner

To give a basis of comparison for the CFLS planner, two baselines were used: the Random Planner and the Simple Prerequisite Planner (SPP). One simulation was run where learners were given LOs from the Random Planner, and in another another simulation learners were given LOs from the SPP. These baseline results are shown in Table 4.1. Two measurements of learner performance were taken for each simulated learner: the score on Final Exam and the % LOs Mastered. The average for each measurement is shown for each aptitude group. In general, and as expected, the low aptitude learners scored lower than the medium aptitude learners who scored lower than the high aptitude learners. This is reflected by the score on Final Exam for both the Random Planner and the SPP. For % LOs Mastered, all learners using the SPP had the same score - all mastering 100% of the LOs. Using the Random planner, high aptitude learners mastered a higher percentage of LOs than the other learners. That high aptitude learners mastered a higher percentage of LOs suggests they were better able to cope with receiving LOs in random order. The low aptitude learners mastered a slightly higher percentage (27.3%) of LOs than the medium aptitude learners (26%). This was not expected and could be attributed to randomness: the low aptitude learners may have been randomly given LOs that were easier than the LOs given randomly to the medium aptitude learners on average.

Table 4.1: Baseline results for each group of simulated learners (high, medium and low aptitude) when visiting LOs randomly and following a simple prerequisite planner.

Planning Type / Aptitude	low	medium	high
Random	N=21	N=26	N=18
Average % LOs Mastered	27.3%	26%	29.2%
Average score on Final Exam (P[learned])	0.107	0.160	0.235
Simple Prerequisite Planner (SPP)	N=21	N=26	N=18
Average % LOs Mastered	100%	100%	100%
Average score on Final Exam (P[learned])	0.619	0.639	0.714

Comparing these two baselines, both performance measurements indicate that learners performed better when using the SPP than when using the Random Planner, for all aptitude groups. That learners performed better with the SPP than Random was indeed expected. Learners would do better when given the LOs in prerequisite order because this is how the evaluation function is written (see Equation 4.1); the evaluation function will give a P[learned] value that is higher if learners have already mastered prerequisite LOs.

Because these baseline results seem to be intuitively plausible, it provides confidence that the simulation is behaving in a reasonable manner. In the next section, I'll discuss the settings of the CFLS planner (i.e. values of b and $f = s$) that led to the highest performance for each aptitude group. I'll then discuss the behaviour of the planner across the different combinations of settings, that is, what happens to learner performance when b is fixed and $f = s$ is changed (Section 4.4.3), and vice versa (Section 4.4.4). Finally, I'll compare the

CFLS planner to the baselines (Section 4.4.5).

4.4.2 CFLS: what combinations of b and $f = s$ worked best?

The best settings for the CFLS planner are those values of b and $f = s$ that led to the highest performance measurements for simulated learners. To try and find the best settings, a total of 25 simulations were run, one for each combination of b , f , and s (where $f = s$) with the values of 1 to 5. The performance measurements for each aptitude group can be seen in Fig. 4.1, which shows the performance over time when $b = 1$ and $f = s = 1$. All the other combinations of b and $f = s$ are included in Appendix B. The orange lines show the average performance measurements for low aptitude learners, green for medium aptitude learners, and blue for high aptitude learners.

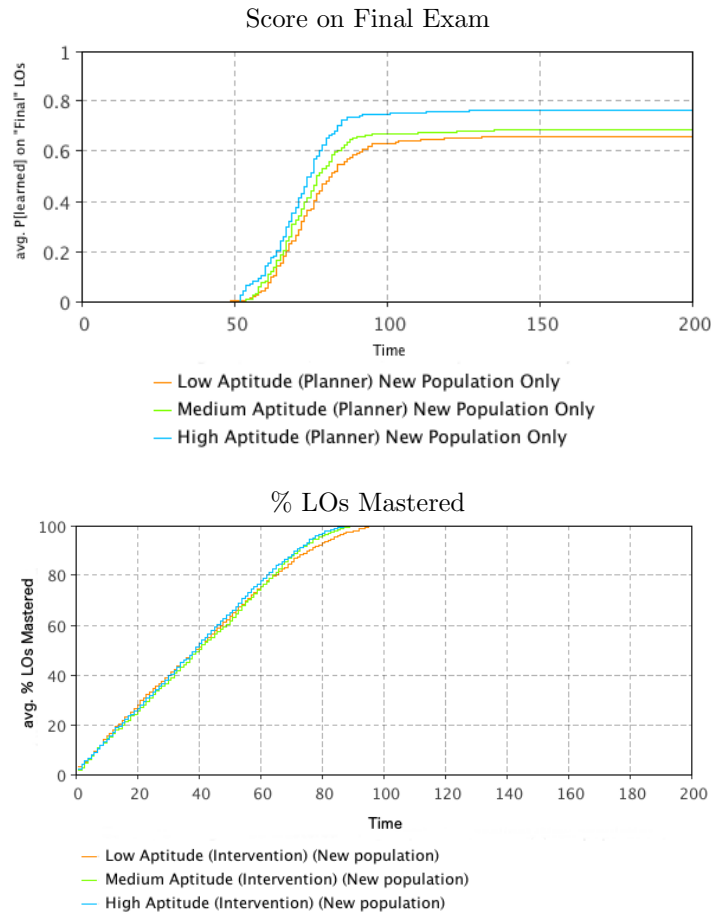


Figure 4.1: Excerpt from Appendix B (Figs. B.1 and B.2)
($b=1$, $f=s=1$)

Across the board, there was not much difference in %LOs Mastered between low, medium and high aptitude learners, although with some combinations of b and $f = s$, the low aptitude learners mastered a much lower percentage of LOs than did medium and high aptitude learners. Looking at the other performance

measurement, score on Final Exam, in general, the low aptitude learners achieved a lower score than did medium learners, who themselves scored lower than high aptitude learners, as expected. The curve shapes for the score on Final Exam (Fig. 4.1, left) are usually more steep than the % LOs Mastered curves (Fig. 4.1, right) because at the start of the simulation the learners would not have interacted with any of the LOs on the Final Exam. Eventually, the learners will interact with those LOs, giving a jump in their score, which can then be increased as the learners interact with more of the Final Exam LOs and even interact with them again to achieve a higher $P[\text{learned}]$ upon seeing the LO a subsequent time. For the rest of this analysis, for both measurements only the values taken at the end of the simulation run (at $t = 200$) are used.

The final values for all 25 simulations are summarized in the heatmaps in Figs. 4.2 and 4.3. Figure 4.2 shows the % LOs mastered, and Fig. 4.3 shows the average score on the Final Exam. Again, measurements are averaged by aptitude group (low, medium, high), giving three heatmaps per measurement, or six in all. In all cases, cells coloured blue indicate a low performance. In Fig. 4.2, blue means there was a low percentage of LOs mastered. In Fig. 4.3, blue means that the simulated learners achieved a low score on the final exam. Cells coloured red indicate a high performance, and white is used for numbers in the middle. Pink means that the number is somewhere between the middle (white) and the very highest performance (dark red). Light blue means the number is somewhere between the middle and the very lowest performance (dark blue). The colour scale from blue to red is applied individually to each aptitude group within the figure to easily pick out the best or worst combinations of b and $f = s$ within each aptitude group.

To compare the heatmaps with the baselines, look at the values in Table 4.1. For example, the table shows that 27.5% is the percentage of LOs Mastered by low aptitude learners using the Random planner. This number can be compared with the values in Fig. 4.2 for low aptitude learners to see that many values are higher than 27.5%, which shows where low aptitude learners mastered a higher percentage of LOs when using the CFLS planner than when using the Random planner.

LOW					MEDIUM					HIGH				
b=1 s=1	b=2 s=1	b=3 s=1	b=4 s=1	b=5 s=1	b=1 s=1	b=2 s=1	b=3 s=1	b=4 s=1	b=5 s=1	b=1 s=1	b=2 s=1	b=3 s=1	b=4 s=1	b=5 s=1
100	21.9	32.5	31	37.7	100	50	45.8	42.2	44.1	100	75	39.3	39.7	41.1
b=1 s=2	b=2 s=2	b=3 s=2	b=4 s=2	b=5 s=2	b=1 s=2	b=2 s=2	b=3 s=2	b=4 s=2	b=5 s=2	b=1 s=2	b=2 s=2	b=3 s=2	b=4 s=2	b=5 s=2
89.6	86	36.9	36.9	40.4	100	100	42.9	40.3	38	100	100	41.7	34.9	40
b=1 s=3	b=2 s=3	b=3 s=3	b=4 s=3	b=5 s=3	b=1 s=3	b=2 s=3	b=3 s=3	b=4 s=3	b=5 s=3	b=1 s=3	b=2 s=3	b=3 s=3	b=4 s=3	b=5 s=3
72.1	68.6	62	43.21	40.1	100	99.4	98.6	42.4	43	100	100	100	42.5	50
b=1 s=4	b=2 s=4	b=3 s=4	b=4 s=4	b=5 s=4	b=1 s=4	b=2 s=4	b=3 s=4	b=4 s=4	b=5 s=4	b=1 s=4	b=2 s=4	b=3 s=4	b=4 s=4	b=5 s=4
77.3	74.4	72.1	66.1	49.3	100	99.3	99.5	99.4	50.8	100	100	100	100	61
b=1 s=5	b=2 s=5	b=3 s=5	b=4 s=5	b=5 s=5	b=1 s=5	b=2 s=5	b=3 s=5	b=4 s=5	b=5 s=5	b=1 s=5	b=2 s=5	b=3 s=5	b=4 s=5	b=5 s=5
68.1	70.7	67.5	67.4	63.5	100	100	100	100	100	100	100	100	100	100

Figure 4.2: Average % Learning Objects Mastered by aptitude group

A general observation is that different aptitude groups found their best performance with differing values of b and $f = s$: the higher the learner's aptitude, the higher values of b and $f = s$ should be used. This can't be seen using Fig. 4.2 (% LOs Mastered) because the performance was the same for all aptitude groups, leaving no stand outs to indicate ideal combinations of b and $f = s$. Instead, use Fig. 4.3 (score on Final Exam) to look for dark red cells. For the low aptitude group, the darkest red appears in the top left corner ($b = 1, f = s = 1$). This means that for low aptitude learners, they received the highest score on the Final

Exam when the CFLS planner used $b = 1$ and $s = 1$. The CFLS planner would match peers based on only having only $b = 1$ most recent learning objects in common, and recommended sequences $f = s = 1$ before re-planning. For medium aptitude learners, the dark red cells occur at slightly higher values of b and s ($b = 1$ or 2 , $s = 2$). The high aptitude learners performed best with higher values still, with the highest average score on Final Exam at $b = 3$ and $s = 3$.

LOW					MEDIUM					HIGH				
b=1 s=1	b=2 s=1	b=3 s=1	b=4 s=1	b=5 s=1	b=1 s=1	b=2 s=1	b=3 s=1	b=4 s=1	b=5 s=1	b=1 s=1	b=2 s=1	b=3 s=1	b=4 s=1	b=5 s=1
0.6587	0.1036	0.1314	0.1283	0.146	0.6894	0.1851	0.2105	0.2099	0.2425	0.7641	0.2514	0.2805	0.2866	0.2702
b=1 s=2	b=2 s=2	b=3 s=2	b=4 s=2	b=5 s=2	b=1 s=2	b=2 s=2	b=3 s=2	b=4 s=2	b=5 s=2	b=1 s=2	b=2 s=2	b=3 s=2	b=4 s=2	b=5 s=2
0.5178	0.4387	0.1398	0.1248	0.1363	0.7004	0.698	0.2058	0.22	0.1972	0.77	0.7694	0.2673	0.2738	0.2748
b=1 s=3	b=2 s=3	b=3 s=3	b=4 s=3	b=5 s=3	b=1 s=3	b=2 s=3	b=3 s=3	b=4 s=3	b=5 s=3	b=1 s=3	b=2 s=3	b=3 s=3	b=4 s=3	b=5 s=3
0.4051	0.266	0.2256	0.1586	0.132	0.6942	0.6761	0.6715	0.1944	0.2152	0.7653	0.7638	0.7727	0.3019	0.3097
b=1 s=4	b=2 s=4	b=3 s=4	b=4 s=4	b=5 s=4	b=1 s=4	b=2 s=4	b=3 s=4	b=4 s=4	b=5 s=4	b=1 s=4	b=2 s=4	b=3 s=4	b=4 s=4	b=5 s=4
0.4138	0.2984	0.3016	0.2755	0.176	0.6931	0.6867	0.6874	0.6856	0.2292	0.768	0.7697	0.7633	0.7697	0.3431
b=1 s=5	b=2 s=5	b=3 s=5	b=4 s=5	b=5 s=5	b=1 s=5	b=2 s=5	b=3 s=5	b=4 s=5	b=5 s=5	b=1 s=5	b=2 s=5	b=3 s=5	b=4 s=5	b=5 s=5
0.357	0.2884	0.2859	0.2679	0.2249	0.6912	0.6884	0.6924	0.6965	0.6899	0.7601	0.7612	0.7591	0.7644	0.7636

Figure 4.3: Average score on Final Exam (P[learned]) by aptitude group

However, there is a limit to the values of b and $f = s$ that should be used. Certain combinations of b and $f = s$ have resulted in a drastic and unexpected drop in performance for all learners. Look along the diagonals where red and blue cells appear right next to each other. This abrupt and consistent drop at $b > s$ was unexpected. Why would there be such a stark contrast between two adjacent cells, for example ($b = 2$, $f = s = 2$) and ($b = 3$, $f = s = 2$)? In both cells, the CFLS planner was recommending sequences 2 LOs long ($f = s = 2$). The difference in b reflects how the CFLS planner was matching peers. When $b = 2$, to find the peers for a target learner, the peers are taken to be all learners who have viewed the same 2 most recent LOs as the target learner (with the same pass/fail result). When $b = 3$, then the peers are taken to be all learners who have viewed the same 3 most recent LOs as the target learner (with the same pass/fail result). One explanation is that there was not enough usage data for the system to consistently find enough peers with the same 3 most recent LOs, thus random matches would have been used instead. The pattern of adjacent red/blue cells with highly different performance measurements appears in a diagonal, creating a red triangle of success, and a blue triangle of failure.

Student's t-test was used to check whether the differences in adjacent cells were statistically significant. For this analysis, it was possible to use paired t-tests because the simulated learners have exactly the same characteristics in all the simulation runs, the only difference being the order in which LOs were interacted with. For example, learner #3 always has **aptitude** = 4, so, there is no difference in that learner between simulation runs. I used a two-tailed t-test because it was not certain whether one distribution was going to be higher or lower than the other.

The t-test was conducted between each adjacent cell in the 5 by 5 heatmap. Each heatmap has 40 possible cell-to-cell comparisons. The comparisons are done for each aptitude group, giving 120 cell-to-cell comparisons in all, each containing the t-test for both performance measurements. These are detailed in Appendix C. An excerpt is included below to discuss within this chapter (Table 4.2). This table shows 12 out of the 120 cell-to-cell comparisons. Numbers in bold are statistically significant (i.e. low p-values). A value of 'n/a'

means that a t-test could not be conducted because the values are the same for both populations (i.e. the simulated learners in each simulation represented by each cell). This happened for the % LOs Mastered when all learners mastered 100% of the LOs.

Table 4.2: P-values for $b = 3$, $f = s = \{1...5\}$
(Excerpt from Appendix C: Table C.8)

Student's t-test	311 vs 322	322 vs 333	333 vs 344	344 vs 355
HIGH APTITUDE	N = 18	N = 18	N = 18	N = 18
p-value (score on Final Exam)	0.4762	2.56×10^{-21}	0.1026	0.3739
p-value (% LOs Mastered)	0.5799	4.30×10^{-15}	n/a	n/a
MEDIUM APTITUDE	N = 26	N = 26	N = 26	N = 26
p-value (score on Final Exam)	0.6404	1.68×10^{-22}	0.1588	0.5483
p-value (% LOs Mastered)	0.1483	2.58×10^{-16}	0.0152	0.0222
LOW APTITUDE	N = 21	N = 21	N = 21	N = 21
p-value (score on Final Exam)	0.5547	0.0156	0.0231	0.6565
p-value (% LOs Mastered)	0.3041	0.0003	0.0004	0.0419

Here's an example of how to cross reference the t-test results in Table 4.2 with the heatmaps. In Table 4.2, there are some bolded numbers in column '333 vs 344' for the % LOs mastered for low and medium aptitude groups. This means the performance was significantly different between learners in two adjacent cells in the heatmap (see low and medium groups in Fig.4.2). Look at the very middle cell ($b = 3$, $s = f = 3$, or '333'), and the cell immediately below ($b = 3$, $s = f = 4$, or '344'). For low aptitude learners, the % LOs mastered was 62% and 72.1% for '333' and '344' respectively (p-value of 0.0004). For medium aptitude learners, the values are 98.6% and 99.5%. Both of these differences are statistically significant, but less so for the medium aptitude learners (p-value of 0.0152). This means that it was statistically better for low and medium aptitude learners to use the CFLS planner with $s = f = 4$ rather than $s = f = 3$, when $b = 3$. In other words, it was better for the CFLS planner to recommend sequences of length $f = s = 4$ rather than $f = s = 3$ when peer neighbourhoods were being formed using $b = 3$ LOs in common. As another example, for high aptitude learners look at the score on Final Exam (Fig. 4.3). For '333' their average score was 0.7727 and for '344' their average score was 0.7633. For high aptitude learners it was actually better to use $s = 3$ rather than $s = 4$, though not significantly so (p=0.1026).

The t-tests verify that the differing values between cells along the red/blue border are statistically significant. To spot the red/blue border cases throughout Appendix C, look for a table having a column with bold values down the whole column, i.e. for all aptitude groups (blue = high, green = medium, orange=low), and for both measurements (score on Final Exam and % LOs Mastered). This can be seen in Table 4.2 in the column named '322 vs 333'. The '322' is an abbreviation for ($b = 3$, $f = 2$, $s = 2$). Because the values are all bolded, it means the simulated learner performance was significantly different between those

two simulations for all learners, for both measurements. These red/blue border cases account for about half of all the cell-to-cell comparisons that showed statistical significance for one or both measurements. There are 50 cell-to-cell comparisons that showed statistical significance, and 24 of these are along the red/blue border. In 23 of these cases, both measurements were significant. In one case, only the % LOs Mastered was significant and the score on Final Exam was not.

One interpretation of these diagonal patterns is that if the planner is recommending sequences of length $f = s = 3$, then the planner must use a base of peers who have no more than 3 LOs in common. Another way to think of this is that if a learner has been matched with a peer group of $b = 2$ LOs then learners absolutely must follow the path ahead for $f = s = 2$ or longer (i.e. always use $s \geq b$) before re-planning occurs. Abandoning the path too soon would certainly be disastrous. To be sure the pattern was real, an extended series of simulations was run. Running $b = 6$ and $s = 5$, it was found that indeed there was a drastic drop in performance. Another row was also run using a fixed $s = 6$ and varying b . Again, a drop in learner performance was found at $b = 7$. The pattern appears to continue on for all values of b and $f = s$.

Outside of the red/blue border, it was rare for both measurements to be statistically significant, but when they were, it was most often with low aptitude learners. There are 26 remaining cell-to-cell comparisons that show statistical significance (that are not along the red/blue border), but only three of these were significant for both measurements. All of these three cases were for the low aptitude group within the red success triangle. Of the 23 cases where only one measurement was significant, 10 were for low aptitude learners, 8 were for medium aptitude learners, and 5 for high aptitude learners.

It wasn't just the score on Final Exam impacting the low aptitude learners. Of the same 23 cases, 15 were significant for % LOs Mastered and 8 were significant for the score on Final Exam. It is perhaps surprising that the % LOs Mastered was significant more often because this measurement deals with all of the LOs in the system. The score on Final Exam only dealt with a small number of LOs and small changes (i.e. an unusual score on one of the LOs in the final exam) would have more impact on this measurement.

4.4.3 Using the same b , did performance increase or decrease with changes to $f = s$?

If the CFLS planner were frozen at a certain value of b , and if settings of b and $f = s$ are chosen within the red success triangle, does the choice of $f = s$ make a difference? For low aptitude learners, it was best not to recommend a long sequence of LOs, regardless of how closely peers were matched. In other words, within the columns within the red success triangles on the heatmaps (i.e. a fixed b value), it was generally best to use lower values for $f = s$, without falling into the blue triangle of failure. There are six cases where low aptitude learners performed worse (where the comparison of one or both performance measurements were statistically significant) when decreasing $f = s$ and keeping the same b . There was only one cases where it was better to use a higher $f = s$: when $b = 3$ it was statistically better to use $f = s = 4$ than $f = s = 3$. This can be seen in the column 333 vs 344 in Table 4.2.

For medium aptitude learners, for $b = 1$ or $b = 2$, it was better to use a higher $f = s = 2$ than to use $f = s = 1$. But, using values of $f = s = 3$ or higher resulted in worsening performance, in the same pattern as low aptitude learners. However, for $b = 3$ and higher, it appeared that using higher values of $f = s$ with the same b resulted in better performance, an opposite pattern to the low aptitude learners.

For high aptitude learners, excluding red/blue border cases, there were no cases where a change in $f = s$ with the same b resulted in statistically different performance.

4.4.4 Using the same $f = s$, did performance increase or decrease with changes to b ?

A pattern one might expect to see is that performance would be higher when the planner uses peers who are the closest match possible, as opposed to using peers with less in common with each other. For the CFLS planner, a better matched peer occurs when a higher b is used; that is, learners who have interacted with more similar LOs with common results. Indeed, increasing b did lead to higher performance in several cases. Looking at Table 4.3, for high aptitude learners, there is a bold value for the score on Final Exam under the column ‘233 vs 333’. This corresponds to the following cells in the heatmaps: ($b = 2, s = 3$) and ($b = 3, s = 3$), where the high aptitude learners average score on Final Exam was higher (0.7638 vs. 0.7727). This means that the score on Final Exam was statistically higher for high aptitude learners when $b = 3$ rather than $b = 2$ when $s = 3$. There are a few other examples where Fig. 4.3 shows high aptitude learners getting higher performance based on using a higher value of b (i.e. 144 and 244; 344 and 444; 155 and 255; 355 and 455), but none are statistically significant.

Table 4.3: P-values for $b = \{1...5\}$, $f = 3$ $s = 3$
(Excerpt from Appendix C: Table C.3)

Student’s t-test	133 vs 233	233 vs 333	333 vs 433	433 vs 533
HIGH APTITUDE	N = 18	N = 18	N = 18	N = 18
p-value (score on Final Exam)	0.7296	0.0172	5.01×10^{-13}	0.8021
p-value (% LOs Mastered)	n/a	n/a	1.95×10^{-13}	0.1498
MEDIUM APTITUDE	N = 26	N = 26	N = 26	N = 26
p-value (score on Final Exam)	0.1086	0.6294	1.29×10^{-22}	0.0593
p-value (% LOs Mastered)	0.0309	0.0169	3.35×10^{-17}	0.8851
LOW APTITUDE	N = 21	N = 21	N = 21	N = 21
p-value (score on Final Exam)	0.0017	0.2648	0.0322	0.0285
p-value (% LOs Mastered)	0.3856	0.0701	0.0029	0.9758

Similarly, there are several cases for medium aptitude learners where it appears learners achieved higher performance for using a higher b , but these were not significant (244 and 344; 255 and 355; 355 and 455). For low aptitude learners, there is only one example (244 and 344). While there is no statistical significance,

it is interesting that the high aptitude have 5 examples, medium aptitude learners have 3 examples, and low aptitude learners have 1 example (244 and 344). This suggests that the higher the aptitude, the more likely learners will benefit from closer matched peers.

In the remaining cases, increasing b did not increase the performance. It seems as though increasing b led to a gradual decline in learner performance. This is especially visible for low aptitude learners in Fig. 4.3: nearly every row starts with high values on the left at $b = 1$, with performance getting worse as b increases. The column ‘133 vs 233’ in Table 4.3 shows the lower value of b resulted in statistically higher score on Final Exam for low aptitude learners, though the % LOs Mastered was not significant.

This can be explained by the fact that the higher the value of b , the more rare a peer will be. A high $b = 5$ means that a peer is defined as someone who has viewed the same most recent 5 LOs with the same pass/fail outcomes. So, it is an expected characteristic of this planner that learner performance will eventually suffer as the values of b get too high for the density of data available. If no peers can be found and random learners are used instead, this could result in a low aptitude learner being recommended a path that worked for a high aptitude learner, which might be too difficult for the low aptitude learner, resulting in lower performance. This also speaks to the issue of data sparsity. The CFLS planner relies on having usage data from other learners. If lots of usage data is available, then the planner might work very well with high values of b . If there is not much data available, then the planner might be forced to keep to low values of b .

4.4.5 How did the CFLS planner compare to the Baselines?

When appropriate values of b and $f = s$ are used (i.e. within the red success triangle), the CFLS planner outperformed the Random planner. No matter what aptitude group or performance measurement was used, all values in both Figs. 4.2 and 4.3 are higher than the corresponding results of the Random planner shown in Table 4.1.

As to the Simple Prerequisite Planner, for low aptitude learners, the CFLS planner only outperformed the SPP in one case, when $b = 1$ and $f = s = 1$. For all other combinations of b and $f = s$, the low aptitude learners using the CFLS planner did not do as well as learners who used the Simple Prerequisite Planner. However, for the medium and high aptitude groups, the CFLS planner outperformed the SPP in all cases within the success triangle. Looking at the score on Final Exam (Fig. 4.3), within the success triangle the high aptitude group achieved higher performance than the SPP at any value of b using $s \geq b$, seeming to do best when $s = b$. Similarly, the medium aptitude group seemed to do well with any value of b within the success triangle, though best results occurred not with $s = b$ but rather $s = b + 1$.

4.5 Conclusion

This experiment was both an exploration of how to tune the CFLS planner, and a test of whether this type of planning would be possible for DOELEs. It is promising on both fronts that most learners performed well

with appropriate values of b and $f = s$. In this section, I discuss the tuning of the CFLS planner and will address broader issues of planning for DOELEs in Chapter 5.

One of the main concerns of instructional planning is how to balance control between the learner and the system. I studied this planning issue by varying the lengths of plans delivered to learners through the CFLS planner. Sometimes, simulated learners were given longer sequences (a higher value of $f = s$), which meant that the system had control for a longer time. Sometimes simulated learners were given shorter sequences (a lower value of $f = s$) which meant the system didn't have control for as long.

The literature suggests that to help them learn, novice learners tend to need more structure, or more system control (such as the expertise reversal effect (Kalyuga et al., 2003 [71])). For the CFLS planner, this means a higher value of $f = s$. In the simulation experiment, one might expect that low aptitude learners should have performed best using higher values of $f = s$. However, the results largely revealed the opposite. Low aptitude learners performed their best using low values of b and $f = s$. Indeed, the simulation results suggest that the higher the aptitude of the simulated learners, the higher the values of b and $f = s$ that should be used.

One explanation is that low aptitude learners were only ever matched with other low aptitude learners. Whenever the CFLS planner created a neighbourhood for a target low aptitude learner, it would draw from the the sequences of LOs of other low aptitude learners. For the sequences to be effective, the neighbourhood size of peers needed to be very large ($b = 1$ means neighbours need only 1 LO in common, thus giving many possible peers), and plans needed to be recalculated after every LO. Perhaps a future version of the CFLS planner would do better if it also looked for good sequences of LOs by allowing medium aptitude learners into the neighbourhood. There may have been a sequence of LOs that a medium aptitude learner succeeded on that would also work for a low aptitude learner.

In addition, high aptitude learners would have been most resilient to receiving recommendations of sequences that were less ideal for them. These learners could follow in the steps of their peers and find great success despite the LOs not being in prerequisite order or the learners never having seen the LOs before. Low aptitude learners would have been more dependent on receiving sequences that were in correct prerequisite order.

This experiment was only a beginning and much remains to be explored to further develop the CFLS planner. In the cycle of designing ALT with simulation, this experiment took the first step to initially narrow down the most promising designs for the CFLS planner. There's still room for further improvement of the CFLS planner. I point out several possibilities for immediate future simulation experiments in the rest of this section.

Future experiments should further investigate the red and blue triangles. Another simulation using $f \neq s$ would allow the designer to check if the triangle shapes are still present, or if a different pattern emerges. This would help to answer the question of whether the drop in average learner performance along the edge of the triangles was because of f or s .

A more advanced CFLS planner would be able to dynamically choose the values of b , f , or s . While it was useful for experimental reasons to use fixed values of b , f , and s within each simulation, this likely isn't the best approach for the real world. These settings could be optimized depending on what is known to work for certain types of learners and what usage data are available. For example, a CFLS planner in the real world might set out to use $b = 3$ but switch on the fly to $b = 2$ if there isn't enough data to find a match.

This experiment showed that the CFLS planner only works for low aptitude learners when low values of b and $f = s$ are used. A designer may want to try to find a way for the CFLS planner to do a better job of helping low aptitude learners while using higher values of b and $f = s$. One approach would be to change the way the CFLS planner creates neighbourhoods, that is, the pool of other potential learners whose paths could be followed. Maybe a low aptitude learner doesn't need to follow the paths of other low aptitude learners so much as they might need to follow in the paths of learners who click the same way they do or have the same goals they do. Or, maybe it was too loose of a restriction for the CFLS planner to allow any permutation of the same b LOs to be a match. Future experiments can check to see what happens if the CFLS planner were to require the exact permutation of b for a match, or whether to require the exact permutation if they are all fails. Answering these question would require more data and would benefit from the use of synthetic data and simulation.

This experiment looked at the issue of how long the system should have control, when it has it. A high value of f and s means the system had control for more LOs. If each LO represents a large course module, then the values of f and s don't need to be very large for the system to have control over a greater portion of content. I didn't deal with the issue of LO granularity in this thesis because of my focus on doing planning without the need to know about the underlying knowledge structure of the LOs. Future simulations could experiment with planning when different grain sizes of LOs are known.

A future simulation could experiment with giving learners control. An actual test of learner control would be to allow the simulated learners to choose learning objects themselves in between the system-selected sequences, whatever their length. Learner control could be emulated by having simulated learners visit random LOs between the system-selected sequences. Such an experiment may require a larger synthetic dataset with more learners and more usage data to allow for peers to find enough sequences in common when they are visiting LOs randomly. It would not be difficult to create the large dataset, but the simulation would take more time to run. No changes would need to be made to the CFLS planner because even learner-chosen sequences of LOs can be used to create neighbourhoods of length b .

When designing future versions of the CFLS planner, it is important to remember that the results, whether they be success triangles or something else, will be a reflection of several things: the evaluation function (Equation 4.1), the threshold of what is considered to be mastered (in this case, $P[\text{learned}] \geq 0.6$), and the collection of learning objects used (i.e. their difficulty levels and what prerequisite and other relationships exist between them). The learners and their attributes would also impact results if learners can influence each other's performance in the way they did in Chapter 3. Finally, the measures used to evaluate

the success or failure of the system are important, as can be seen by the different results obtained with the two measures used in the experiment, and by the clustering of learners into three aptitude groups.

In this experiment, there was not much difference in results between the high and medium aptitude learners. This is due to the combination of the weights chosen for the evaluation function (e.g. how heavily aptitude was weighted), and the choice of $P[\text{learned}] > 0.6$ as the threshold to consider a LO as mastered, and the nature of the prerequisite graph that was used. Questions that remain unanswered include: Would the success triangles have looked the same if different weights or dimensions were used? How would learners of various aptitudes be affected by a change to the threshold for mastery than > 0.6 ? To tune the weights of the evaluation function, it may have helped to break down the aptitude groups into six: very-high, high, medium-high, medium-low, low, and very-low.

Because of the seemingly endless options, it is easy to become overwhelmed by the possibilities of simulation. Grounding the simulated learners, LOs or their behaviours in the real world will help narrow down the possibilities. After the initial experiment like the one described in this chapter, the next step in the cycle of designing ALT with simulation is to compare the simulation results with the real world.

For example, instead of an arbitrary prerequisite graph of simulated LOs, the designer could substitute representations of a real course, where a course is a collection of LOs in the ecological approach. Instead of using synthetic data from simulated learners, the usage data from real learners can be imported in a simulation to check if it would be sufficient to launch a CFLS planner. The designer could check if the sequences of LOs being recommended by the CFLS planner actually seem to make sense. Eventually, the CFLS planner would need to be used with real learners. The usage data from those learners would likely reveal things that were not anticipated with simulation. A designer can check if the red and blue triangles still appear when the CFLS planner is used with real learners. The designer might discover they need to change the evaluation function, or they might want to change the behaviour of the CFLS planner. After changing the simulation model to address the gaps, the designer can run the simulation again to refine their new design. Then the improved design can be tried in the real world, continuing the cycle to better serve real learners.

Simulation is a much needed tool for the design of ALT. The simulation experiment in this chapter has provided insight into the promise of the CFLS planner for environments like DOELES.

CHAPTER 5

CONCLUSION

The most powerful support tools available for learners today are a result of ALT research. Using simulation and the ecological approach, I studied two main problems that need to be addressed for ALT to run in DOELEs: how to explore pedagogical issues and how to explore ALT system design issues.

I studied the pedagogical issues of how peers can impact each other's learning in a changing environment, and of how to choose a sequence of LOs. It would have been possible to do simulation studies of these issues without the EA. But by using the EA, the discoveries made are applicable to DOELEs. The observations in Chapter 3 about peer impact apply to DOELEs, where learners could be coming and going. And the tuning exercise of the CFLS planner in Chapter 4 can be readily extended to different DOELEs where there are different simulated learners and different numbers and types of LOs.

I also studied the system design issues of how to tune the CFLS planner and how ALT can provide personalized support to learners without the need for a large amount of knowledge engineering. I further showed how simulation can be used as part of the ALT system designer's toolkit to shed light on important issues and make predictions for how a new design might behave under various conditions.

In Section 5.1, I discuss the bigger picture of instructional planning (IP) for DOELEs and relate this to the current active research area of automatically discovering prerequisite graphs from usage data. I then reflect on how the predictions made in these simulation experiments can relate to the real world in Section 5.2.

5.1 Instructional Planning for DOELEs

A DOELE is a place where the material to be learned is constantly changing, and learners are also coming and going with changing goals. Most ALT is not designed for DOELEs because most ALT relies on extensive knowledge engineering to create underlying knowledge and control structures to support it. Knowledge engineering is inconsistent with DOELEs because in such dynamic environments it is impractical to continuously re-engineer these structures with each change.

I used the ecological approach (EA) because it is an architecture that synchronizes well with DOELEs. In the EA, there is no centralized data structure that needs to be built beforehand, and no re-engineering or metadata markup is required when the learners or LOs change. Instead, each time a learner interacts with a LO, information that is known about the learner and the results of the interaction are captured and

stored with the LO. While my experiments didn't go so far as to add or remove LOs during the simulations, changing the LOs could be readily done in future work. The only requirement for adding a new LO is that it gets used by someone, anyone, and the outcome is recorded. The more usage data captured, the better the system will be able to match a LO to its appropriate niche.

This architecture enables the CFLS planner to work in DOELEs. By interacting with LOs, the simulated learners created usage data, which provided all the information that the CFLS planner needed. The CFLS planner's goal is to select a sequence of LOs that would be ideally suited for a target learner. To select the LOs, the CFLS planner made no attempt to inventory the concepts mastered by the learner, nor did it have any knowledge of the underlying prerequisite structure of the LOs. Instead, the CFLS planner directed the learner to follow sequences of LOs that worked for other learners who had an experience similar to the target learner's most recent experience. The CFLS planner drew upon the $P[\text{learned}]$ values that had been stored with previous learner-to-LO interactions to form the peer neighbourhoods and to find the paths learners followed next. In this way, the CFLS planner provided personalized support without needing to know the details of the material being learned.

The results are a promising sign that planning based on collaborative filtering is attainable for DOELEs. In one baseline, simulated learners visited LOs randomly. These learners were unable to master more than an average of 30% of the LOs, nor did they achieve a passing score on the final exam. In contrast, simulated learners using the CFLS planner that had been configured with the right settings (of b and $f = s$), were able to master 100% of the LOs. In some cases, the learners using the CFLS planner achieved even higher scores on the final exam than simulated learners using the other baseline, the Simple Prerequisite Planner (SPP), which was built using a knowledge engineering approach.

The CFLS planner was launched from usage data gathered from learners who had used the SPP. The usage data enabled the CFLS planner to recommend paths based on the experience of other learners who had already consumed LOs in prerequisite order. The CFLS planner would have also worked had it been launched using data that had been gathered by learners interacting with LOs randomly, but it would have required much more data. Learners would have had to first randomly stumble upon sequences of LOs that worked before those sequences would be recognized as successful and then recommended to other learners. The CFLS planner was successful because it leveraged the results of the SPP, but its true strength is that the sequences were found without needing to know the actual underlying prerequisite relationships. Using simulation, developers can test the CFLS planner with many different datasets to identify when a given set of usage data is sufficient to launch the CFLS planner.

The CFLS planner is a contribution to IP because it is the only known instructional planner that is based entirely on usage data and not on a centralized data structure. This thesis has also contributed to ALT research by demonstrating that the ecological approach can support intelligent tutoring behaviour. The CFLS planner showed how to select the next learning object in a DOELE, where relatively little is known about the learners or LOs.

The related topic of how to automatically discover a prerequisite structure from learner behaviour is an active area of research. It is widely agreed that creating a prerequisite graph is laborious for ALT researchers and that an automated solution is needed (Brunskill, 2011 [18], Vuong et al., 2011 [143]). For similar reasons, methods have also been developed to automatically generate concept maps (Tseng et al., 2005 [133]). Another use for an automatically discovered prerequisite structure is for validating a structure manually created by human experts who may have conflicting opinions.

A well-known machine learning technique to do this is Association Rule Mining (ARM). ARM infers a prerequisite relationship exists between two LOs when it finds in the usage data from many learners that they were significantly more likely to master one LO if they had previously mastered the other. One challenge with ARM is that the machine learning algorithms often return too many rules for the system designer to make sense of them. Improvements can be made on the quality of the rules discovered by allowing the system designer to become interactively involved in the rule discovery process using their knowledge of the task domain (Marinica & Guillet, 2010 [97]).

Research into automatically discovering the prerequisite graph is related to my research in that it shares the motivation of wanting to avoid knowledge engineering. But ARM is often conducted within closed courses where knowledge engineering of the task domain has already been completed. When ARM is combined with a task domain ontology, it is called Semantic Data Mining (Dou et al. [38]). In DOELEs, there is no task domain ontology to enable the prerequisite graph to be discovered.

Rather than try to discover a prerequisite graph, the CFLS planner didn't need one at all. The CFLS planner was able to check which sequences worked and which ones didn't, depending on what it discovered about the learner and their recent activity from the EA usage data. Planning based on usage data opens up possibilities for new bottom-up approaches to instructional planning intended for large DOELEs.

In the real world, many more factors beyond prerequisites would influence the best ordering of the content for a learner. Whether or not a prerequisite relationship even exists between two LOs could vary from learner to learner because of their different experiences and attributes. The notion of prerequisite usually considers only content relationships among material. Other factors include learning specialized knowledge before generalized (or vice versa, depending on the learner), or learning concrete before abstract (Wong and Looi, 2009 [147]) (or vice versa), or learning things in an order that is based on a learning style (Franzoni & Assar, 2009 [50]). The CFLS planner could detect new ideal orderings of topics that are due to other attributes of the learners or LOs, not just prerequisites.

Perhaps the CFLS planner would find that learners from different disciplines tend to do better when approaching the content a certain way. Or perhaps the CFLS planner would discover a certain unusual ordering of LOs that works really well for a small cluster of learners. In my experiment, the CFLS planner would have had to discover paths in prerequisite order because this is the only structure that would have been implicit in the synthetic usage data. The usage data would have reflected the prerequisite relationships because these were built into the evaluation function. But in the real world, the usage data would reflect a

multitude of other factors that impact the learner’s success, and the CFLS planner could detect those factors at work.

Different learners could effectively have a unique ideal implicit ‘prerequisite graph’ that reflects much more than prerequisite relationships. The planner doesn’t need to understand why, only that the ordering had worked for learners similar to a certain profile. The CFLS planner demonstrated that the EA allows learning paths to be discovered that may have never even been thought of by a human designer.

5.2 Constructive Simulations of ALT

Some might argue that simulation can’t help further the understanding of pedagogical issues because there is not enough known about human learning, personality types, or social issues for an accurate simulation to be possible. But I’ve argued that the simulation doesn’t have to model all the details of the real world to be useful. Simulation permits investigators to discard unnecessary detail to advance their understanding, especially in early phases of research, before conducting an expensive study. When designing the simulation model, I modelled learners, LOs, and their interactions only to the extent needed for the study, and nothing more.

To study peer impact, I didn’t need a full understanding of the psychology of peer influence on learning. Rather, I created two models of how a learner could either benefit from or be held back by how they are doing relative to their peers (*attracted-to-Peer-score* and *repelled-from-Peer-score*). I gave each simulated learner one of these types, and made it a part of the evaluation function, which determined P[learned], a number to represent whether the learner learned the LO.

To study the pedagogical issue of what to do next, I needed to capture the idea that prerequisites are real. Human teachers know that real learners need to understand certain basics before more advanced concepts can be grasped. In the simulation, I captured the notion of the need for basic knowledge by creating a prerequisite graph for the LOs (and ensuring that the CFLS planner would have no knowledge of it). The evaluation function was built so that P[learned] would reflect whether the learner had mastered the prerequisites. The resulting P[learned] values were enough to power the CFLS planner.

Each experiment provided insight into the pedagogical issue being studied that I likely never would have found without simulation. In the peer impact experiment, I discovered certain circumstances led to a group of low aptitude learners actually performing better on average than a group of high aptitude learners. This happened for *repelled-from-Peer-score* learners during the hard mode simulations when new simulated learners were introduced. Wording this as a hypothesis for the real world, it would be: If a real learner is affected by their peers in a manner that matches the model of *repelled-from-Peer-score* learners, and if this real learner is in a DOELE that is dense with difficult material, and if new learners are introduced, then the following can be expected. If they are a high aptitude learner, they will struggle a lot in the immediate future and perform much more poorly than they would have had if the new learners had not joined. If they are a low

aptitude learner, they will not suffer from the disruption as much as everyone else.

Two aspects of the peer impact experiment need further exploration. First, it is important to check to see if real learners ever actually behave like the *repelled-from-Peer-score* or *attracted-to-Peer-score* learners (and if not, use different models). Second, if these models are reflected in real behaviour, it is important to develop ways of supporting learners by highlighting (or minimizing) peer influences that would benefit (or be harmful to) them. Such support could take the form of a set of guidelines such as, ‘In a DOELE that is dense with difficult material, and if new learners are introduced, then for high-aptitude, *repelled-from-Peer-score* learners you should highlight more examples of peer success stories until they have recovered from the environmental shift of the new learners joining’.

In the CFLS planner experiment, I found that the number of steps to plan ahead (s and f) is related to the number of previous steps that the target learner has in common with the peers used to create the neighbourhood (b). I also found that for low aptitude learners, the CFLS planner only worked when configured with low values of b and $f = s$. These observations provided me with valuable feedback to improve the CFLS planner in future experiments, such as to try to allow low aptitude learners to follow the paths of successful medium aptitude learners. These results lead to other questions for future simulations to explore. For example, do the patterns hold up with different LOs or a different population of learners? What would happen if last year’s learners took this year’s course? Indeed, simulation may be the only way to thoroughly test theories of peer impact and planners like the CFLS planner for DOELEs. A true benefit of simulation is the ability to try the CFLS planner with PEER-IMPACT in the evaluation function. For example, the CFLS planner could find different paths for *attracted-to-Peer-score* vs. *repelled-from-Peer-score* learners.

Simulation results are a prediction. But, would the predictions uncovered in this thesis be useful to anyone in the real world? A human teacher might not be able to do anything with knowledge about low aptitude learners doing better than high aptitude learners in a very specific situation. And, this is only one example. A simulation has potential to provide a great many predictions for a great many situations. These predictions may be too numerous and most of them too rare for a human teacher to recognize and make use of this knowledge.

In general, simulation predictions would be most useful for ALT researchers. ALT researchers take the predictions from simulations and incorporate them into an ALT system. The system can then intervene appropriately to help learners when particular situations arise, either on its own, or by summoning a human teacher and informing them of the nature of the situation facing the learner.

In addition, such predictions could help analyze real world usage data, such as why a particular learner may have struggled. For example, the usage data from a real learner could be compared to see if they fit as a *repelled-from-Peer-score* type or *attracted-to-Peer-score* type. Knowing a learner’s type combined with knowing conditions of the learning environment at the time (i.e. the types of learners and LOs in the DOELE at the time) may help reveal the reasons behind their struggles.

Simulation allows ALT researchers to try many hypotheses, rejecting many and pursuing only the most

promising ones. I was able to test the feasibility of the CFLS planner for DOELEs and provided information that could help focus a real world study in the future. For instance, a future study of the CFLS planner should pay attention to the success triangles, and may not need to focus on the combinations of b and $f = s$ within the triangles where the learner performance wasn't significantly different in the simulation.

Ultimately, learning technology designed with the help of simulation needs to work in the real world. In later iterations of a study, there will be times when the simulation's predictions need to be more accurate. A more realistic evaluation function would produce values of $P[\text{learned}]$ that are more similar to the $P[\text{learned}]$ values produced by real learners. For a more accurate prediction for a specific DOELE, the LOs in the simulation should be made to correspond to LOs in the real world (similar number of LOs, similar difficulty level, and prerequisite relationships). In the real world, human learners don't have a single number that represents their aptitude nor do they necessarily have a style of peer impact that can reliably predict how they will be impacted by their peers. However, real people do have different backgrounds, privileges and ways that they process information and make decisions that are unique to them. ALT researchers can create models, such as the models of peer impact, and initialize the simulated learners with attributes that are more similar to a cohort of real learners. In this way, the behaviour of the real learners can be compared to that of the simulated learners, allowing researchers to refine and improve the peer impact model, or whatever else is being studied with the help of simulation.

As ALT researchers begin to adopt simulation, they join scientists in other fields who gain from its benefits. The social scientist Robert Axelrod in a 1997 article (Axelrod, 1997 [7]) explained that simulation is a new form of scientific thinking that joins the fundamentals of induction and deduction. Scientists use induction to discover patterns within data. Scientists use deduction to draw conclusions out of a starting set of axioms. The third form of scientific thinking, simulation, starts with a set of axioms that run their course, producing data along the way, data from which patterns can be discovered. Because it is often so difficult to predict the results of a simulation, Axelrod says, "simulation is often the only viable way to study populations of agents who are adaptive rather than fully rational."

This thesis has shown that by having a controlled environment, it is possible to further explore interesting pedagogical and system design issues using simulation. Simulation can complement existing methodologies as a valuable addition to the toolkit for ALT researchers.

REFERENCES

- [1] IEEE Standard for Learning Object Metadata. *IEEE Std. 1484.12.1-2002*, 2002.
- [2] Sophiana Chua Abdullah and Roger E Cooley. Using Simulated Students to Evaluate an Adaptive Testing System. *Proceedings of the International Conference on Computers in Education ICCE*, 1:614–618, 2002.
- [3] Marwan Al-Tawil, Dhavalkumar Thakker, and Vania Dimitrova. Nudging to Expand User’s Domain Knowledge while Exploring Linked Data. In *Proceedings of the 3rd International Workshop on Intelligent Exploration of Semantic Data co-located with the 13th International Semantic Web Conference (ISWC 2014)*, pages 54–65. CEUR-WS.org, October 2014.
- [4] Jonathan K Alt. *Learning from Noisy and Delayed Rewards: The Value of Reinforcement Learning to Defense Modeling and Simulation*. PhD thesis, Naval Postgraduate School, Monterey, California, September 2012.
- [5] Jennitta Andrea. Generative Acceptance Testing for Difficult-to-Test Software. *Extreme Programming and Agile Processes in Software Engineering*, LNCS 3092:29–37, 2004.
- [6] Brian A.A. Antao, Arthur J Brodersen, John R Bourne, and Jeffrey R Cantwell. Building Intelligent Tutorial Systems for Teaching Simulation in Engineering Education. *IEEE Transactions on Education*, 35(1):50–56, 1992.
- [7] Robert Axelrod. Advancing the Art of Simulation in the Social Sciences. *Complexity*, 3(2):16–22, 1997.
- [8] Álvaro Barbero, Mario Salvador González-Rodríguez, Juan de Lara, and Manuel Alfonseca. Multi-Agent Simulation of an Educational Collaborative Web System. *European Simulation and Modelling Conference*, 2007.
- [9] Amy L Baylor and Yanghee Kim. Simulating Instructional Roles through Pedagogical Agents. *International Journal of Artificial Intelligence in Education*, 15(2):95–115, 2005.
- [10] Stephen Blessing, Stephen B Gilbert, Steven Ourada, and Steven Ritter. Authoring Model-Tracing Cognitive Tutors. *International Journal of Artificial Intelligence in Education*, 19(2):189–210, 2009.
- [11] Benjamin S. Bloom. *Taxonomy of Educational Objectives, Cognitive and Affective Domains*. Longman Group United Kingdom, June 1969.
- [12] Lars Bollen and Wouter R van Joolingen. SimSketch: Multiagent Simulations Based on Learner-Created Sketches for Early Science Education. *IEEE Transactions on Learning Technologies*, 6(3):208–216, 2013.
- [13] Paul Bradley. The History of Simulation in Medical Education and Possible Future Directions. *Medical Education*, 40(3):254–262, March 2006.
- [14] Steffen Brauer and Thomas C Schmidt. Group Formation in eLearning-enabled Online Social Networks. In *ICL’12: Proceedings of the 15th International Conference on Interactive Collaborative Learning*. IEEE, 2012.
- [15] Javier Bravo and Alvaro Ortigosa. Validating the Evaluation of Adaptive Systems by User Profile Simulation. In Stephan Weibelzahl and Alexandra Cristea, editors, *Proceedings of Workshop held at the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006)*, pages 479–483, Dublin, Ireland, June 2006.

- [16] Barbara Brecht. *Determining the Focus of Instruction : Content Planning for Intelligent Tutoring Systems*. PhD thesis, University of Saskatchewan, Saskatoon, Saskatchewan, 1990.
- [17] Christopher A Brooks, Jim E Greer, Erica Melis, and Carsten Ullrich. Combining ITS and eLearning Technologies: Opportunities and Challenges. *Intelligent Tutoring Systems*, LNCS 4053:278–287, 2006.
- [18] Emma Brunskill. Estimating Prerequisite Structure From Noisy Data. In M Pechenizkiy, T Calders, C Conati, S Ventura, C Romero, and J Stamper, editors, *Proceedings of the 4th International Conference on Educational Data Mining*, pages 217–222, Eindhoven, the Netherlands, 2011.
- [19] Peter Brusilovsky and Nicola Henze. Open Corpus Adaptive Educational Hypermedia. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, pages 671–696. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [20] Peter Brusilovsky and Julita Vassileva. Course Sequencing Techniques for Large-Scale Webbased Education. *International Journal of Continuing Engineering Education and Lifelong Learning*, 13:75–94, 2003.
- [21] Susan Bull, Andrew Mabbott, and Abdallatif S Abu-Issa. UMPTEEN: Named and Anonymous Learner Model Access for Instructors and Peers. *International Journal of Artificial Intelligence in Education*, 17(3):227–253, 2007.
- [22] Robin Burke. The Wasabi Personal Shopper: A Case-Based Recommender System. In *Proceedings of the 11th International Conference on Innovative Applications of Artificial Intelligence*, 1999.
- [23] Robin Burke. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, November 2002.
- [24] Sílvia César Cazella, Eliseo Berni Reategui, and Patricia Alejandra Behar. Recommendation of Learning Objects Applying Collaborative Filtering and Competencies. In Nicholas Reynolds and Márta Turcsányi-Szabó, editors, *Key Competencies in the Knowledge Society*, pages 35–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [25] Karina L Cela, Miguel Ángel Sicilia, and Salvador Sánchez. Social Network Analysis in E-Learning Environments: A Preliminary Systematic Review. *Educational Psychology Review*, 27(1):219–246, July 2014.
- [26] John Champaign. *Peer-Based Intelligent Tutoring Systems: A Corpus-Oriented Approach*. PhD thesis, University of Waterloo, 2012.
- [27] John Champaign and Robin Cohen. A model for content sequencing in intelligent tutoring systems based on the ecological approach and its validation through simulated students. In *FLAIRS Conference*, 2010.
- [28] Tak-Wai Chan. Integration-Kid: A Learning Companion System. *Proceedings of the 12th International Conference on Artificial Intelligence*, 2:1094–1099, 1991.
- [29] Geraldine Clarebout and Jan Elen. Advice on Tool Use in Open Learning Environments. *Journal of Educational Multimedia and Hypermedia*, 17(1):81–97, 2008.
- [30] Gemma Corbalan, Liesbeth Kester, and Jeroen J G Van Merriënboer. Towards a personalized task selection model with shared instructional control. *Instructional Science*, 34(5):399–422, September 2006.
- [31] Albert Corbett, Linda Kauffman, Ben Maclaren, Angela Wagner, and Elizabeth Jones. A Cognitive Tutor for Genetics Problem Solving: Learning Gains and Student Modeling. *Journal of Educational Computing Research*, 42(2):219–239, February 2010.
- [32] Geoff Cumming and Anne McDougall. Mainstreaming AIED into Education? *International Journal of Artificial Intelligence in Education*, 11:197–207, June 2000.

- [33] Ben K Daniel, Gord McCalla, and Richard A Schwier. Social Network Analysis Techniques: Implications for Information and Knowledge Sharing in Virtual Learning Communities. *International Journal of Advanced Media and Communication*, 2(1):20–34, 2008.
- [34] Michel C Desmarais and Ildiko Pelczer. On the Faithfulness of Simulated Student Performance Data. In Ryan S Baker, Agathe Merceron, and Philip I Pavlik, editors, *Proceedings of the 3rd International Conference on Educational Data Mining*, Pittsburgh PA, 2010.
- [35] Elizabeth Diener and Nelda Hobbs. Simulating Care: Technology-Mediated Learning in Twenty-First Century Nursing Education. *Nursing Forum*, 47(1):34–38, January 2012.
- [36] Nianbo Dong and Mark W Lipsey. Biases in Estimating Treatment Effects Due to Attrition in Randomized Controlled Trials and Cluster Randomized Controlled Trials: A Simulation Study. In *Proceedings of the Spring Research Conference of the Society for Research on Educational Effectiveness*, Washington, DC, 2011.
- [37] Fabiano Dorça. Implementation and use of Simulated Students for Test and Validation of new Adaptive Educational Systems: a Practical Insight. *International Journal of Artificial Intelligence in Education*, 25:319–345, February 2015.
- [38] Dejing Dou, Hao Wang, and Haishan Liu. Semantic Data Mining: A Survey of Ontology-based Approaches. *Proceedings of the IEEE th International Conference on Semantic Computing IEEE ICSC*, pages 244–251, 2015.
- [39] Martin Dougiamas. Moodle open-source learning platform. <http://www.moodle.org>. Accessed: 2014-07-05.
- [40] Hendrik Drachsler, Hans Hummel, and Rob Koper. Using Simulations to Evaluate the Effects of Recommender Systems for Learners in Informal Learning Networks. In *Proceedings of the 2nd Social Information Retrieval for Technology Enhanced Learning SIRTEL*, Maastricht, Netherlands, 2008.
- [41] Hendrik Drachsler, Hans Hummel, Bert van den Berg, Jannes Eshuis, Waterink Wim, Rob Nadolski, Adriana Berlanga, Nanda Boers, and Rob Koper. Effects of the ISIS Recommender System for navigation support in self-organised Learning Networks. *Journal of Educational Technology and Society*, 12(3):122–135, 2009.
- [42] Benedict du Boulay and Rosemary Luckin. Modelling Human Teaching Tactics and Strategies for Tutoring Systems. *International Journal of Artificial Intelligence in Education*, 12(3):393–404, 2001.
- [43] Pragya Dwivedi and Kamal K. Bharadwaj. E-Learning Recommender System for a Group of Learners Based on the Unified Learner Profile Approach. *Expert Systems*, 32(2):264–276, 2015.
- [44] Michael Eagle, Matthew Johnson, Tiffany Barnes, and Acey Boyce. Exploring Player Behavior with Visual Analytics. In *International Conference on the Foundations of Digital Games*, pages 380–383, Chania, Crete, Greece, 2013.
- [45] Jon A Elorriaga and I Fernández-Castro. Using Case-Based Reasoning in Instructional Planning. Towards a Hybrid Self-improving Instructional Planner. *International Journal of Artificial Intelligence in Education*, 11(4):416–449, 2000.
- [46] Graham Erickson, Stephanie Frost, Scott Bateman, and Gord McCalla. Using the Ecological Approach to Create Simulations of Learning Environments. In H Chad Lane, Kalina Yacef, Jack Mostow, and Philip Pavlik, editors, *Proceedings of the 16th International Conference on Artificial Intelligence in Education*, pages 411–420, Memphis, TN, April 2013.
- [47] Robert G Farrell, Soyini D Liburd, and John C Thomas. Dynamic Assembly of Learning Objects. In *Proceedings of the 13th International World Wide Web Conference*. ACM, May 2004.
- [48] Carlena Cochi Ficano. Peer effects in college academic outcomes – Gender matters! *Economics of Education Review*, 31(6):1102–1115, December 2012.

- [49] Gigi Foster. It's not your peers, and it's not your friends: Some progress toward understanding the educational peer effect mechanism. *Journal of Public Economics*, 90(8-9):1455–1475, September 2006.
- [50] Ana Lidia Franzoni and Saïd Assar. Student Learning Styles Adaptation Method Based on Teaching Strategies and Electronic Media. *Educational Technology & Society*, 12(4):15–29, 2009.
- [51] Norm Friesen. The International Learning Object Metadata Survey. *International Review of Research in Open and Distance Learning*, 5(3), November 2004.
- [52] Stephanie Frost and Gord McCalla. An Approach to Developing Instructional Planners for Dynamic Open-Ended Learning Environments. In Jesus Boticario and Kasia Muldner, editors, *Proceedings of the Workshops at the 17th International Conference on Artificial Intelligence in Education AIED*, Madrid, Spain, 2015.
- [53] Vasco Furtado and Eurico Vasconcelos. Geosimulation in Education: A System for Teaching Police Resource Allocation. *International Journal of Artificial Intelligence in Education*, 17(1), 2007.
- [54] Antonio Garrido and Eva Onaindia. Assembling Learning Objects for Personalized Learning: An AI Planning Perspective. *IEEE Intelligent Systems*, 28(2):64–73, 2013.
- [55] G. David Garson. Computerized Simulation in the Social Sciences: A Survey and Evaluation. *Simulation & Gaming*, 40(2):267–279, May 2008.
- [56] Andreas Gegenfurtner, Carla Quesada-Pallarès, and Maximilian Knogler. Digital simulation-based training: A meta-analysis. *British Journal of Educational Technology*, 45(6):1097–1114, July 2014.
- [57] Nigel Gilbert and Klaus Troitzsch. *Simulation For The Social Scientist*. McGraw-Hill Education (UK), February 2005.
- [58] Daniel V Girzadas Jr, Lamont Clay, Jennifer Caris, Kathleen Rzechula, and Robert Harwood. High fidelity simulation can discriminate between novice and experienced residents when assessing competency in patient care. *Medical Teacher*, 29(5):472–476, January 2007.
- [59] Arthur C Graesser, Patrick Chipman, Brian C Haynes, and Andrew Olney. AutoTutor: An Intelligent Tutoring System With Mixed-Initiative Dialogue. *IEEE Transactions on Education*, 48(4):612–618, November 2005.
- [60] Pierre-Paul Grassé. La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes sociaux*, 6(1):41–80, 1959.
- [61] Jim Greer, Gord McCalla, Julita Vassileva, and Ralph Deters. Lessons Learned in Deploying a Multi-Agent Learning Support System: The I-Help Experience. In *Proceedings of AI in Education AIED*, pages 410–421, 2001.
- [62] Hicham Hage and Esma Aïmeur. Harnessing Learner's Collective Intelligence: A Web2.0 Approach to E-Learning. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, pages 438–447, Montreal, Canada, 2008. Springer Berlin Heidelberg.
- [63] Michael Hannafin. Learning in Open-Ended Environments: Assumptions, Methods and Implications. *Educational Technology*, 34(8):48–55, 1994.
- [64] Eric A Hanushek, John F Kain, Jacob M Markman, and Steven G Rivkin. Does Peer Ability Affect Student Achievement? *Journal of Applied Econometrics*, 18(5):527–544, 2003.
- [65] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems TOIS*, 22(1):5–53, January 2004.

- [66] Roya Hosseini, I-Han Hsiao, Julio Guerra, and Peter Brusilovsky. What Should I Do Next? Adaptive Sequencing in the Context of Open Social Student Modeling. In Gráinne Conole, Tomaž Klobučar, Christoph Rensing, Johannes Konert, and Élise Lavoué, editors, *Design for Teaching and Learning in a Networked World*, pages 155–168. Springer International Publishing, 2015.
- [67] Ana Iglesias, Paloma Martínez, Ricardo Aler, and Fernando Fernández. Learning teaching strategies in an Adaptive and Intelligent Educational System through Reinforcement Learning. *Applied Intelligence*, 31(1):89–106, August 2009.
- [68] Judith Israel and Robert M Aiken. Supporting Collaborative Learning With An Intelligent Web-Based System. *International Journal of Artificial Intelligence in Education*, 26(1):365–377, March 2007.
- [69] W. Lewis Johnson, Jeff W. Rickel, and James C. Lester. Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education*, 11:47–78, 2000.
- [70] Johnson, W. Lewis. Serious Use of a Serious Game for Language Learning. *International Journal of Artificial Intelligence in Education*, 20(2):175–195, January 2010.
- [71] Slava Kalyuga, Paul L Ayres, Paul Chandler, and John Sweller. The Expertise Reversal Effect. *Educational Psychologist*, 38(1):23–31, 2003.
- [72] Stan Karanasios, Dhavalkumar Thakker, Lydia Lau, David Allen, Vania Dimitrova, and Alistair Norman. Making sense of digital traces: An activity theory driven ontological approach. *Journal of the American Society for Information Science and Technology*, 64(12):2452–2467, September 2013.
- [73] Abbey C Karich, Matthew K Burns, and Kathrin E Maki. Updated Meta-Analysis of Learner Control Within Educational Technology. *Review of Educational Research*, 84(3):392–410, July 2014.
- [74] Judy Kay, Peter Reimann, Elliot Diebold, and Bob Kummerfeld. MOOCs: so many learners, so much potential.... *Intelligent Systems, IEEE*, 28(3):70–77, August 2013.
- [75] Kathleen Keogh and Liz Sonenberg. Designing for Planned Emergence in Multi-agent Systems. In Aditya Ghose, Nir Oren, Pankaj Telang, and John Thangarajah, editors, *Coordination, Organizations, Institutions and Norms in Agent Systems X*, pages 97–113. Educational Data Mining, 2014.
- [76] Ali Abdul Khaliq, Maurizio Di Rocco, and Alessandro Saffiotti. Stigmergic algorithms for multiple minimalistic robots on an RFID floor. *Swarm Intelligence*, 8(3):199–225, July 2014.
- [77] John S Kinnebrew and Gautam Biswas. Identifying Learning Behaviors by Contextualizing Differential Sequence Mining with Action Features and Performance Evolution. In *Proceedings of the 5th International Conference on Educational Data Mining*, pages 57–64, Chania, Greece, 2012.
- [78] John S Kinnebrew, Kirk M Loretz, and Gautam Biswas. A Contextualized, Differential Sequence Mining Method to Derive Students’ Learning Behavior Patterns. *JEDM - Journal of Educational Data Mining*, 5(1):190–219, May 2013.
- [79] John S Kinnebrew, James R Segedy, and Gautam Biswas. Analyzing the temporal evolution of students’ behaviors in open-ended learning environments. *Metacognition and Learning*, 9(2):187–215, March 2014.
- [80] David Kiss. The impact of peer achievement and peer heterogeneity on own achievement growth: Evidence from school transitions. *Economics of Education Review*, 37:58–65, December 2013.
- [81] Kenneth R Koedinger, Ryan S.J.d. Baker, Kyle Cunningham, Alida Skogsholm, Brett Leber, and John Stamper. A Data Repository for the EDM community: The PSLC DataShop. In Gautam Biswas, Susan Bull, Judy Kay, and Antonija Mitrovic, editors, *Proceedings of the 15th International Conference on Artificial Intelligence in Education*, Auckland, New Zealand, January 2013. Springer Berlin Heidelberg.
- [82] Rob Koper. Increasing Learner Retention in a Simulated Learning Network Using Indirect Social Interaction. *Journal of Artificial Societies and Social Simulation*, 8(2), 2005.

- [83] Steeve Laberge, Tony Lenihan, Sima Shabani, and Fuhua Lin. Multiagent Coordination for Planning and Enacting an Assessment Game. In *Workshop on MultiAgent System Based Learning Environments at the Conference on Intelligent Tutoring Systems 2014*, Honolulu, Hawaii, 2014.
- [84] Susan M. Land. Cognitive Requirements for Learning with Open-Ended Learning Environments. *Educational Technology Research and Development*, 48(3):61–78, September 2000.
- [85] Kimberly A Lawless and Scott W Brown. Multimedia learning environments: Issues of learner control and navigation. *Instructional Science*, 25(2):117–131, March 1997.
- [86] Advanced Distributed Learning. SCORM 2004 4th Edition. Sponsored by the Office of the Under Secretary of Defense for Personnel and Readiness, United States Government. <http://www.adlnet.gov/scorm/scorm-2004-4th/> Accessed August 2, 2014.
- [87] Advanced Distributed Learning. Training & Learning Architecture (TLA): Project Tin Can. Sponsored by the Office of the Under Secretary of Defense for Personnel and Readiness, United States Government. <http://www.adlnet.gov/tla/tin-can/> Accessed August 2, 2014.
- [88] June Lee. Effectiveness of Computer-based Instructional Simulation: a Meta Analysis. *International Journal of Instructional Media*, 26(1):71–85, 1999.
- [89] Krittaya Leelawong and Gautam Biswas. Designing Learning by Teaching Agents: The Betty’s Brain System. volume 18, pages 181–208, August 2008.
- [90] Xu Lin. Identifying Peer Effects in Student Academic Achievement by Spatial Autoregressive Models with Group Unobservables. *Journal of Labor Economics*, 28(4):825–860, October 2010.
- [91] Margaret L Loper. *Modeling and Simulation in the Systems Engineering Life Cycle*. Simulation Foundations, Methods and Applications. Springer London, London, 2015.
- [92] Fabian Lorig, Nils Dammenhayn, David-Johannes Müller, and Ingo J Timm. Measuring and Comparing Scalability of Agent-Based Simulation Frameworks. In Jörg P Müller, Wolf Ketter, Gal Kaminka, Gerd Wagner, and Nils Bulling, editors, *Behavior and Social Computing*, pages 42–60. Springer International Publishing, December 2015.
- [93] René Mandiau, Alexis Champion, Jean-Michel Auberlet, Stéphane Espié, and Christophe Kolski. Behaviour based on decision matrices for a coordination between agents in a urban traffic simulation. *Applied Intelligence*, 28(2):121–138, 2008.
- [94] Nikos Manouselis, Hendrik Drachsler, Riina Vuorikari, Hans Hummel, and Rob Koper. Recommender Systems in Technology Enhanced Learning. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B Kantor, editors, *Recommender Systems Handbook*, pages 387–415. Springer US, Boston, MA, 2011.
- [95] Yan Mao, Julita Vassileva, and Winfried Grassmann. A System Dynamics Approach to Study Virtual Communities. In *Proceedings of the 40th Hawaii International Conference on System Sciences*, pages 178–178a. IEEE, 2007.
- [96] Nikki J Maran and Ronnie J Glavin. Low- to high-fidelity simulation - a continuum of medical education? *Medical Education*, 37(s1):22–28, November 2003.
- [97] Claudia Marinica and Fabrice Guillet. Knowledge-Based Interactive Postmining of Association Rules Using Ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 22(6):784–797, April 2010.
- [98] Noboru Matsuda, William W Cohen, Jonathan Sewall, Gustavo Lacerda, and Kenneth R Koedinger. Predicting Students’ Performance with SimStudent: Learning Cognitive Skills from Observation. In *Proceeding of the 2007 conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*. IOS Press, June 2007.

- [99] Noboru Matsuda and Kurt VanLehn. Decision Theoretic Instructional Planner for Intelligent Tutoring Systems. In *Proceedings of Workshop on Modeling Human Teaching Tactics and Strategies at the Intelligent Tutoring Systems Conference at the 5th International Conference on Intelligent Tutoring Systems (ITS'00)*, pages 72–83, 2000.
- [100] Gord McCalla. The Search for Adaptability, Flexibility, and Individualization: Approaches to Curriculum in Intelligent Tutoring Systems. In Marlene Jones and Philip H Winne, editors, *Adaptive Learning Environments*, pages 91–121. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.
- [101] Gord McCalla. The Ecological Approach to the Design of E-Learning Environments: Purpose-based Capture and Use of Information About Learners. *Journal of Interactive Media in Education*, 7:1–23, 2004.
- [102] Gord McCalla, Larry Reid, and Peter Schneider. Plan creation, plan execution and knowledge acquisition in a dynamic microworld. *International Journal of Man-Machine Studies*, 16:89–112, January 1980.
- [103] Ben Medler and Brian Magerko. Scribe: A Tool for Authoring Event Driven Interactive Drama. In Stefan Göbel, Rainer Malkewitz, and Ido Iurgel, editors, *Third International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, pages 139–150, 2006.
- [104] Joseph S Mertz. Using A Simulated Student for Instructional Design. *International Journal of Artificial Intelligence in Education*, 8:116–141, 1997.
- [105] Ruth Meyer. Event-Driven Multi-agent Simulation. In *International Workshop on Multi-Agent-Based Simulation XV*, Paris, France, 2015.
- [106] L D Miller, Leen-Kiat Soh, Ashok Samal, and Gwen Nugent. iLOG: A Framework for Automatic Annotation of Learning Objects with Empirical Usage Metadata. *International Journal of Artificial Intelligence in Education*, 21(3):215–236, January 2012.
- [107] Antonija Mitrovic and Stellan Ohlsson. Evaluation of a Constraint-Based Tutor for a Database Language. *International Journal of Artificial Intelligence in Education (IJAIED)*, 10:238–256.
- [108] Fatma Mizikaci. A systems approach to program evaluation model for quality in higher education. *Quality Assurance in Education*, 2006.
- [109] Olivier Motelet and Nelson Baloian. Hybrid System for Generating Learning Object Metadata. In *Sixth IEEE International Conference on Advanced Learning Technologies*, pages 563–567, Kerkrade, Netherlands, May 2006. IEEE.
- [110] Richard T Murphy and Lola Rhea Appel. Evaluation of the PLATO IV Computer-based Education System in the Community College. *ACM SIGCUE Outlook*, 12(1), January 1978.
- [111] Rob Nadolski, Bert V den Berg, Adriana Berlanga, Hendrik Drachsler, Hans Hummel, Rob Koper, and Peter Sloep. Simulating light-weight Personalised Recommender Systems in Learning Networks: A case for Pedagogy-Oriented and Rating-based Hybrid Recommendation Strategies. *Journal of Artificial Societies and Social Simulation*, 12(1), 2008.
- [112] Sayooran Nagulendra and Julita Vassileva. Understanding and controlling the filter bubble through interactive visualization: a user study. In *Proceedings of the 25th ACM conference on Hypertext and social media*, pages 107–115, 2014.
- [113] Walter Nicholson and Frank Westhoff. General Equilibrium Models: Improving the Microeconomics Classroom. *The Journal of Economic Education*, 40(3):297–314, 2009.
- [114] Sabrina Nusrat and Julita Vassileva. Simulating a trust-based service recommender system for decentralised user modelling environment. *International Journal of Trust Management in Computing and Communications*, 1(2):121–139, 2013.

- [115] Xavier Ochoa and Erik Duval. Towards Automatic Evaluation of Learning Object Metadata Quality. In John F Roddick, V Richard Benjamins, Samira Si-said Cherfi, Roger Chiang, Christophe Claramunt, Ramez A Elmasri, Fabio Grandi, Hyoil Han, Martin Hepp, Miltiadis D Lytras, Vojislav B Mišić, Geert Poels, Il-Yeol Song, Juan Trujillo, and Christelle Vangenot, editors, *Proceedings of ER Workshops*, pages 372–381, Tucson, Arizona, USA, 2006. Springer Berlin Heidelberg.
- [116] Fatemeh Orooji, Hessam S Sarjoughian, and Fattaneh Taghiyareh. Modeling & Simulation of Educational Multi-Agent Systems in DEVS-Suite. In *Proceedings of the 5th International Symposium on Telecommunications*, pages 956–961. IEEE, 2010.
- [117] Marc-Andre Orthmann, Martin Friedrich, Uwe Kirschenmann, Katja Niemann, Maren Scheffel, Hans-Christian Schmitz, and Martin Wolpers. Usage-based Clustering of Learning Objects for Recommendation. *Proceedings of the 11th IEEE International Conference on Advanced Learning Technologies*, pages 553–557, April 2011.
- [118] Darwyn R. Peachey and Gordon I. McCalla. Using Planning Techniques in Intelligent Tutoring Systems. *International Journal of Man-Machine Studies*, 24(1):77–98, 1986.
- [119] Reinhard Pekrun, Thomas Goetz, Anne C Frenzel, Petra Barchfeld, and Raymond P Perry. Measuring emotions in students learning and performance: The Achievement Emotions Questionnaire (AEQ). *Contemporary Educational Psychology*, 36(1):36–48, January 2011.
- [120] David V Pynadath and Stacy Marsella. PsychSim: Modeling Theory of Mind with Decision-Theoretic Agents. . In *Proceedings of the 19th International joint conference on Artificial Intelligence*, pages 1181–1186, 2005.
- [121] Ashok Ranchhod, Călin Gurău, Euripides Loukis, and Rohit Trivedi. Evaluating the educational effectiveness of simulation games: A value generation model. *Information Sciences*, 264(C):75–90, April 2014.
- [122] Alessandro Ricci, Andrea Omicini, Mirko Viroli, Luca Gardelli, and Enrico Oliva. Cognitive Stigmergy: Towards a Framework Based on Agents and Artifacts. In D Weyns, H V D Parunak, and F Michel, editors, *E4MAS’06: Proceedings of the 3rd international conference on Environments for Multi-Agent Systems*, pages 124–140. University of Bologna, Springer-Verlag, 2007.
- [123] James M Richards. A Simulation Study of the Use of Change Measures to Compare Educational Programs. *American Educational Research Journal*, 12(3):299–311, July 1975.
- [124] Steven Ritter, John R Anderson, Kenneth R Koedinger, and Albert Corbett. Cognitive Tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review*, 14(2):249–255, 2007.
- [125] Devshri Roy, Sudeshna Sarkar, and Sujoy Ghose. Automatic Extraction of Pedagogic Metadata from Learning Content. *International Journal of Artificial Intelligence in Education*, 18(2):97–118, 2008.
- [126] Ad Schellekens, Fred Paas, Alexander Verbraeck, and Jeroen J G Van Merriënboer. Flexible programmes in higher professional education: expert validation of a flexible educational model. *Innovations in Education and Teaching International*, 47(3):283–294, August 2010.
- [127] Li-ping Shen Shen and Rui-min. Learning Content Recommendation Service Based-On Simple Sequencing Specification. *Advances in Web-Based Learning-ICWL 2004*, 2004.
- [128] E Christine Shin, Diane L Schallert, and Wilhelmina C Savenye. Effects of Learner Control, Advice, and Prior Knowledge on Young Students’ Learning in a Hypertext Environment. *Educational Technology Research and Development*, 42(1):33–46, January 1994.
- [129] George Siemens, Pierre Dillenbourg, Gerhard Fischer, Danielle McNamara, and Nikol Rummel. Where are the Learning Sciences in the MOOC debate ? Technical report, 2014.
- [130] Ali Simsek. Learner Control. In *Encyclopedia of the Sciences of Learning*, pages 1718–1721. Springer US, Boston, MA, 2012.

- [131] Tiffany Tang and Gord McCalla. Utilizing Artificial Learners to Help Overcome the Cold-Start Problem in a Pedagogically-Oriented Paper Recommendation System. In *Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 245–254, Eindhoven, The Netherlands, January 2004.
- [132] XJ Technologies. Anylogic simulation software tool, accessed July 2014. <http://www.xjtek.com/anylogic/>.
- [133] Shian-Shyong Tseng, Pei-Chi Sue, Jun-Ming Su, Jui-Feng Weng, and Wen-Nung Tsai. A new approach for constructing the concept map. *Computers & Education*, 49(3):691–707, November 2007.
- [134] Joshua Underwood and Rosemary Luckin. What is AIED and why does Education need it? Technical report, 2011.
- [135] Kurt VanLehn. The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education*, 16(3):227–265, 2006.
- [136] Kurt VanLehn, Stellan Ohlsson, and Rod Nason. Applications of Simulated Students: An Exploration. *Journal of Artificial Intelligence in Education*, 5:135–175, 1996.
- [137] Julita Vassileva. Reactive Instructional Planning to Support Interacting Teaching Strategies. *Proceedings of the 7-th World Conference on AI and Education*, 1995.
- [138] Julita Vassileva and Lingling Sun. Using Community Visualization to Stimulate Participation in Online Communities. *e-Service Journal*, 6(1):3–39, October 2007.
- [139] Julita Vassileva and Barbara Wasson. Instructional Planning Approaches: From Tutoring Towards Free Learning. In *Proceedings of Euro-AIED*, pages 1–8, Lisbon, Portugal, 1996.
- [140] Udeep Tandukar Vassileva and Julita. Ensuring Relevant and Serendipitous Information Flow in Decentralized Online Social Network. *Proceedings of the 15th International Conference on Artificial Intelligence Methodology, Systems and Applications*, pages 79–88, August 2012.
- [141] Maria Virvou, Konstantinos Manos, and George Katsionis. An evaluation agent that simulates students’ behaviour in intelligent tutoring systems. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 4872–4877, 2003.
- [142] Aurora Vizcaíno. A Simulated Student Can Improve Collaborative Learning. *International Journal of Artificial Intelligence in Education*, 15(1):3–40, 2005.
- [143] Annalies Vuong, Tristan Nixon, and Brendon Towle. A Method for Finding Prerequisites Within a Curriculum. In M Pechenizkiy, T Calders, C Conati, S Ventura, C Romero, and J Stamper, editors, *Proceedings of the 4th International Conference on Educational Data Mining*, pages 211–216, Eindhoven, Netherlands, 2011.
- [144] World Wide Web Consortium (W3C). Semantic web standards, accessed february 2017. <https://www.w3.org/standards/semanticweb/>.
- [145] Barbara Wasson. Instructional Planning and Contemporary Theories of Learning: Is this a Self-Contradiction? In *Proceedings of the European Conference on Artificial Intelligence in Education*, 1996.
- [146] Kathryn R. Wentzel and Kathryn Caldwell. Friendships, Peer Acceptance, and Group Membership: Relations to Academic Achievement in Middle School. *Child Development*, 68(6):1198–1209, December 1997.
- [147] Lung-Hsiang Wong and Chee-Kit Looi. Adaptable Learning Pathway Generation with Ant Colony Optimization. *Educational Technology & Society*, 12(3):309–326, 2009.

- [148] Osman Yaşar. Computational Math, Science, and Technology: A New Pedagogical Approach to Math and Science Education. In *Proceedings of the International Conference on Computational Science and Its Applications, Pt*, pages 807–816, Assisi, Italy.
- [149] Yijun Yu, Huaizhong Lin, and Chun Chen. Personalized Web Recommendation Based on Path Clustering. In *Proceedings of the 7th International Conference on Flexible Query Answering Systems*, pages 148–154, Milan, Italy, 2006. Springer Berlin Heidelberg.
- [150] Yiwen Zhang and M Tanniru. An Agent-Based Approach to Study Virtual Learning Communities. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 11–11c. IEEE, 2005.
- [151] Yu Zhang, Jiang Wu, and Lizhu Ma. A Multi-Agent School Simulation Based on Hierarchical Social Networks. *Intelligent Information Management*, 6(4):196–210, 2014.
- [152] Yuqi Zhang and Jian Cao. Personalized Recommendation Based on Behavior Sequence Similarity Measures. In Longbing Cao, Hiroshi Motoda, Jaideep Srivastava, Ee-Peng Lim, Irwin King, Philip S Yu, Wolfgang Nejdl, Guandong Xu, Gang Li, and Ya Zhang, editors, *Behavior and Social Computing*, pages 165–177. Springer International Publishing, 2013.
- [153] Qin Zhao and Tracy Linderholm. Anchoring effects on prospective and retrospective metacomprehension judgments as a function of peer performance information. *Metacognition and Learning*, 6(1):25–43, December 2010.
- [154] Baoyao Zhou, Siu Cheung Hui, and Kuiyu Chang. An Intelligent Recommender System using Sequential Web Access Patterns. In *IEEE Conference on Cybernetics and Intelligent Systems*, pages 393–398. IEEE, 2004.

APPENDIX A

PEER IMPACT SIMULATIONS

Appendix A shows the results from the peer impact simulations of Chapter 3 for the very interested reader. There were six experimental conditions, listed below. Because of the randomness, each condition was run six times, giving a total of thirty-six graphs. For compactness, two graphs are printed per line.

fiftyfifty, hard mode In this condition, 50% of the simulated learners have the *attracted-to-Peer-score* personality, and 50% have the *repelled-from-Peer-score* personality. This condition was run in hard mode, which means the learning objects had High **difficulty level** and the learners had Low **aptitude**.

fiftyfifty, easy mode In this condition, 50% of the simulated learners have the *attracted-to-Peer-score* personality, and 50% have the *repelled-from-Peer-score* personality. This condition was run in easy mode, which means the learning objects had Low **difficulty level** and the learners had High **aptitude**.

attracted-to-peer-score, hard mode In this condition, 80% of the simulated learners have the *attracted-to-Peer-score* personality, and 20% have the *repelled-from-Peer-score* personality. This condition was run in hard mode, which means the learning objects had High **difficulty level** and the learners had Low **aptitude**.

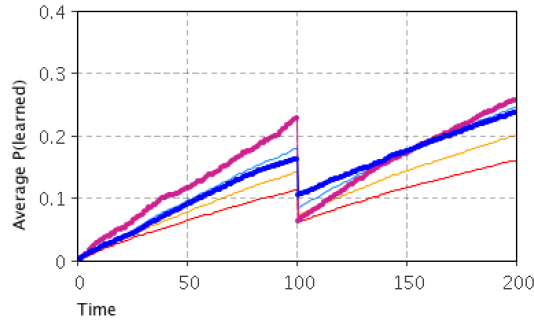
attracted-to-peer-score, easy mode In this condition, 80% of the simulated learners have the *attracted-to-Peer-score* personality, and 20% have the *repelled-from-Peer-score* personality. This condition was run in easy mode, which means the learning objects had Low **difficulty level** and the learners had High **aptitude**.

repelled-from-Peer-score, hard mode In this condition, 20% of the simulated learners have the *repelled-from-Peer-score* personality, and 80% have the *repelled-from-Peer-score* personality. This condition was run in hard mode, which means the learning objects had High **difficulty level** and the learners had Low **aptitude**.

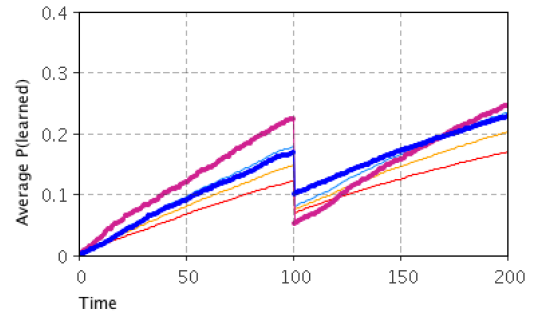
repelled-from-Peer-score, easy mode In this condition, 20% of the simulated learners have the *repelled-from-Peer-score* personality, and 80% have the *repelled-from-Peer-score* personality. This condition was run in easy mode, which means the learning objects had Low **difficulty level** and the learners had High **aptitude**.

- Class Average
- P(learned) – attracted-to-peer-score
- P(learned) – repelled-from-peer-score
- P(learned) – repelled-from-peer-score (HA)
- P(learned) – repelled-from-peer-score (LA)

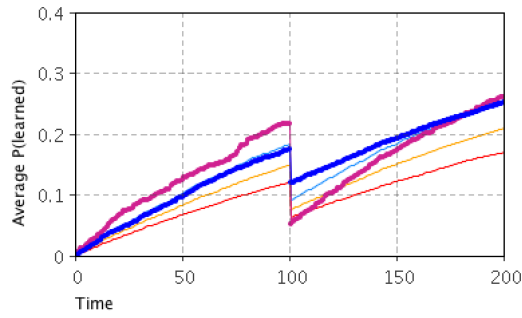
Legend



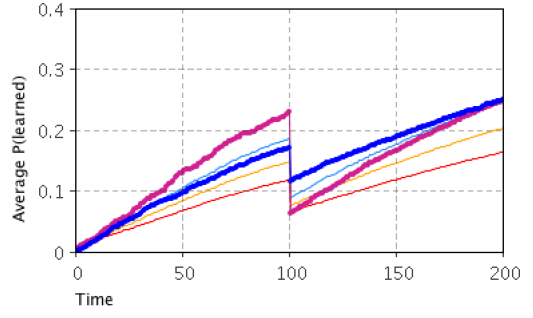
simulation 1



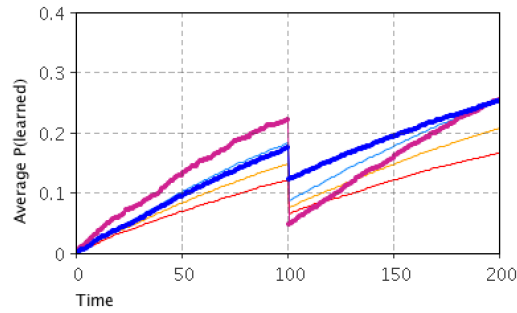
simulation 2



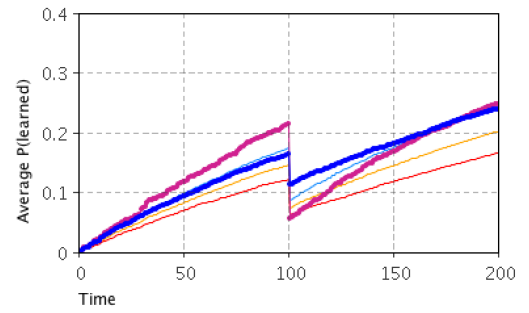
simulation 3



simulation 4



simulation 5



simulation 6

Figure A.1: fiftyfifty, hard mode

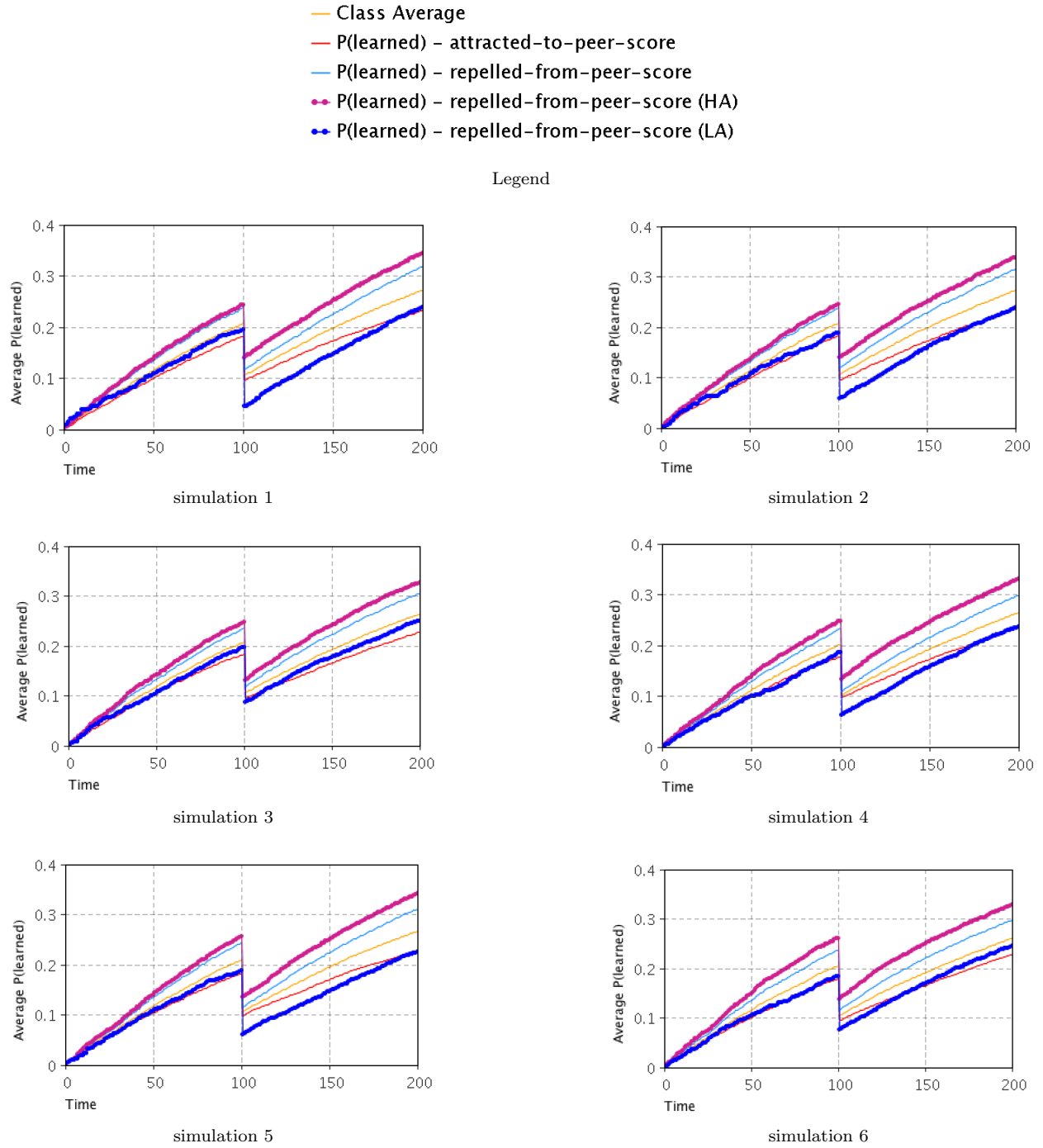
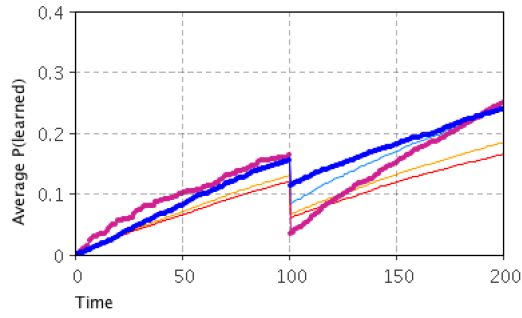


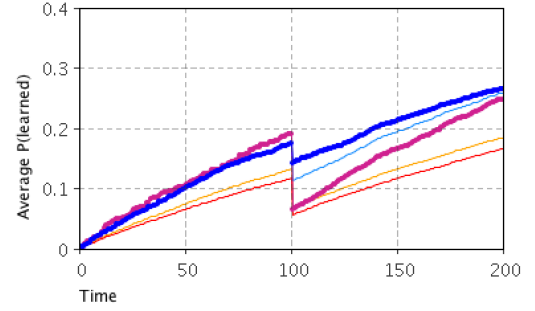
Figure A.2: fiftyfifty, easy mode

- Class Average
- P(learned) – attracted-to-peer-score
- P(learned) – repelled-from-peer-score
- P(learned) – repelled-from-peer-score (HA)
- P(learned) – repelled-from-peer-score (LA)

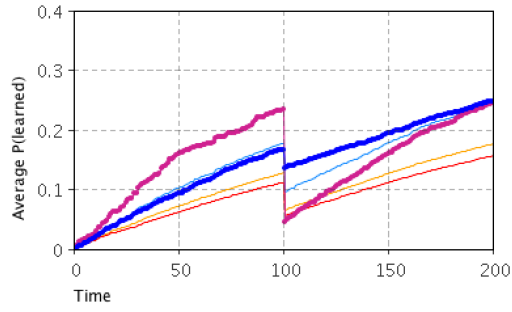
Legend



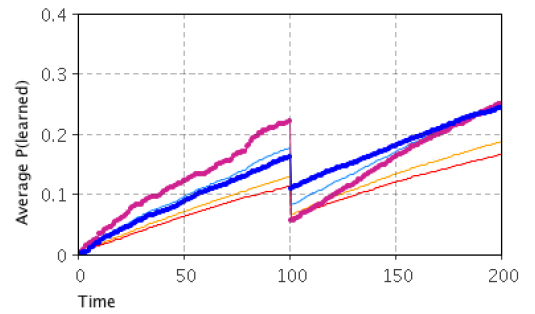
simulation 1



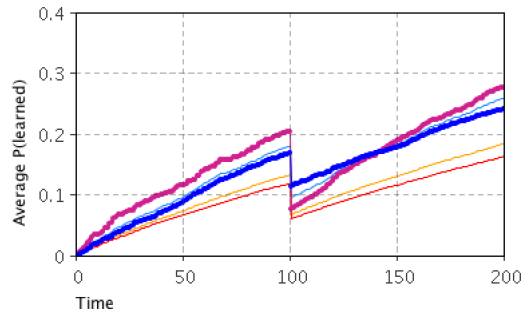
simulation 2



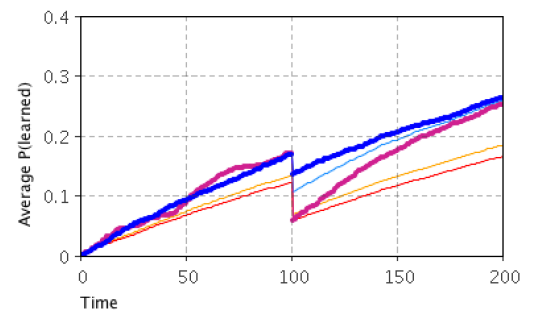
simulation 3



simulation 4



simulation 5

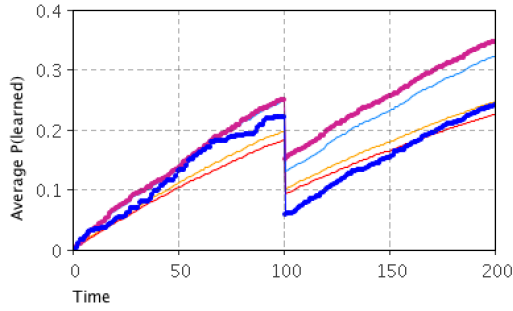


simulation 6

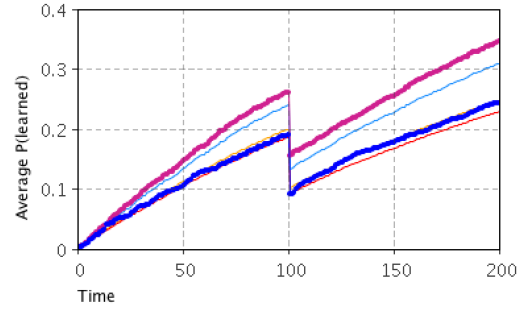
Figure A.3: mostlyattracted, hard mode

- Class Average
- P(learned) – attracted-to-peer-score
- P(learned) – repelled-from-peer-score
- P(learned) – repelled-from-peer-score (HA)
- P(learned) – repelled-from-peer-score (LA)

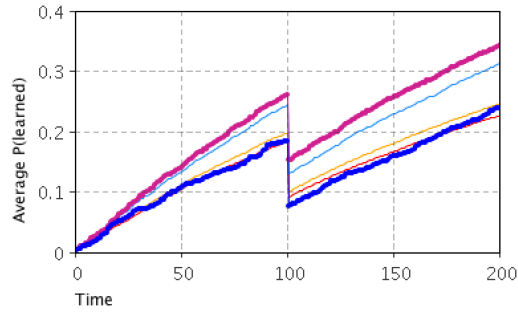
Legend



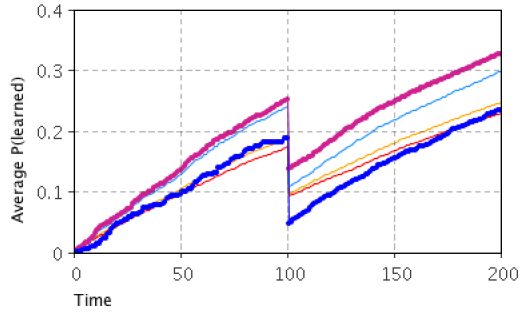
simulation 1



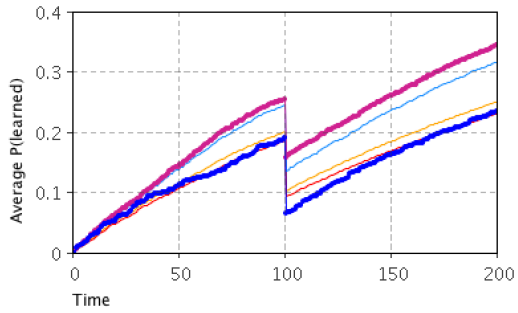
simulation 2



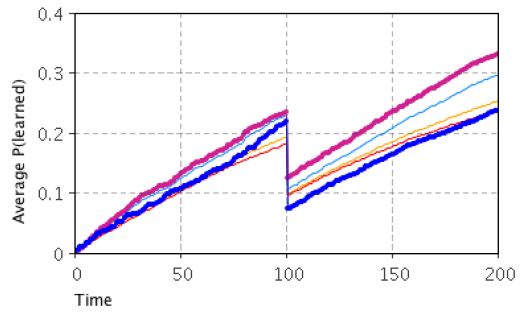
simulation 3



simulation 4



simulation 5

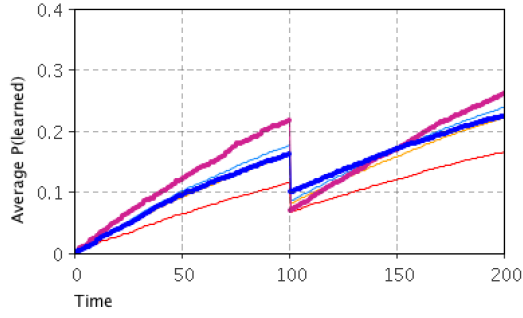


simulation 6

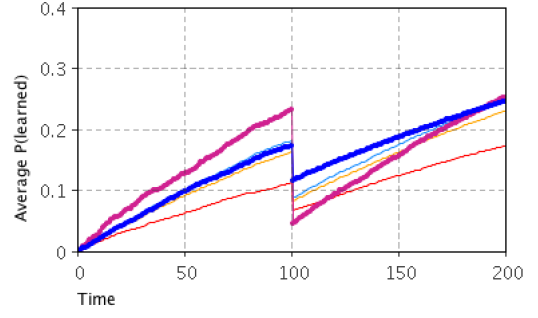
Figure A.4: mostlyattracted, easy mode

- Class Average
- P(learned) – attracted-to-peer-score
- P(learned) – repelled-from-peer-score
- P(learned) – repelled-from-peer-score (HA)
- P(learned) – repelled-from-peer-score (LA)

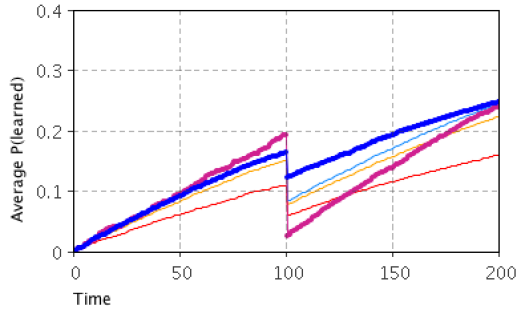
Legend



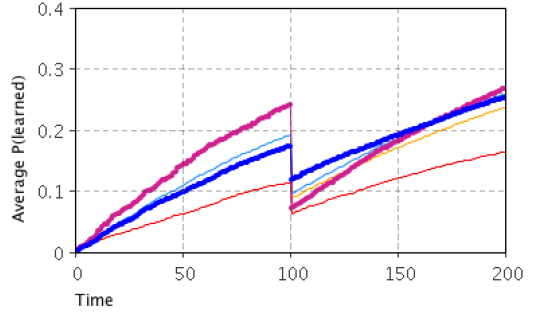
simulation 1



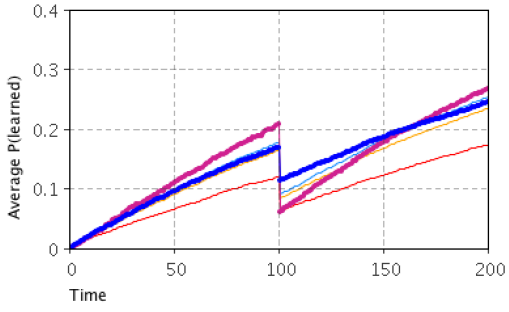
simulation 2



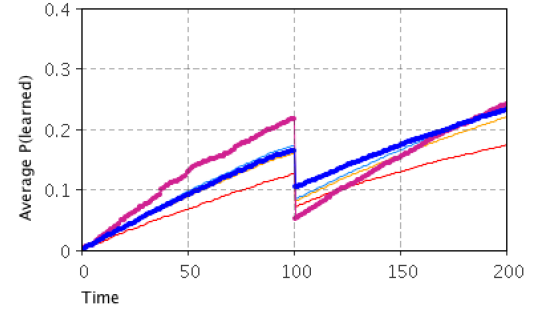
simulation 3



simulation 4



simulation 5



simulation 6

Figure A.5: mostlyrepelled, hard mode

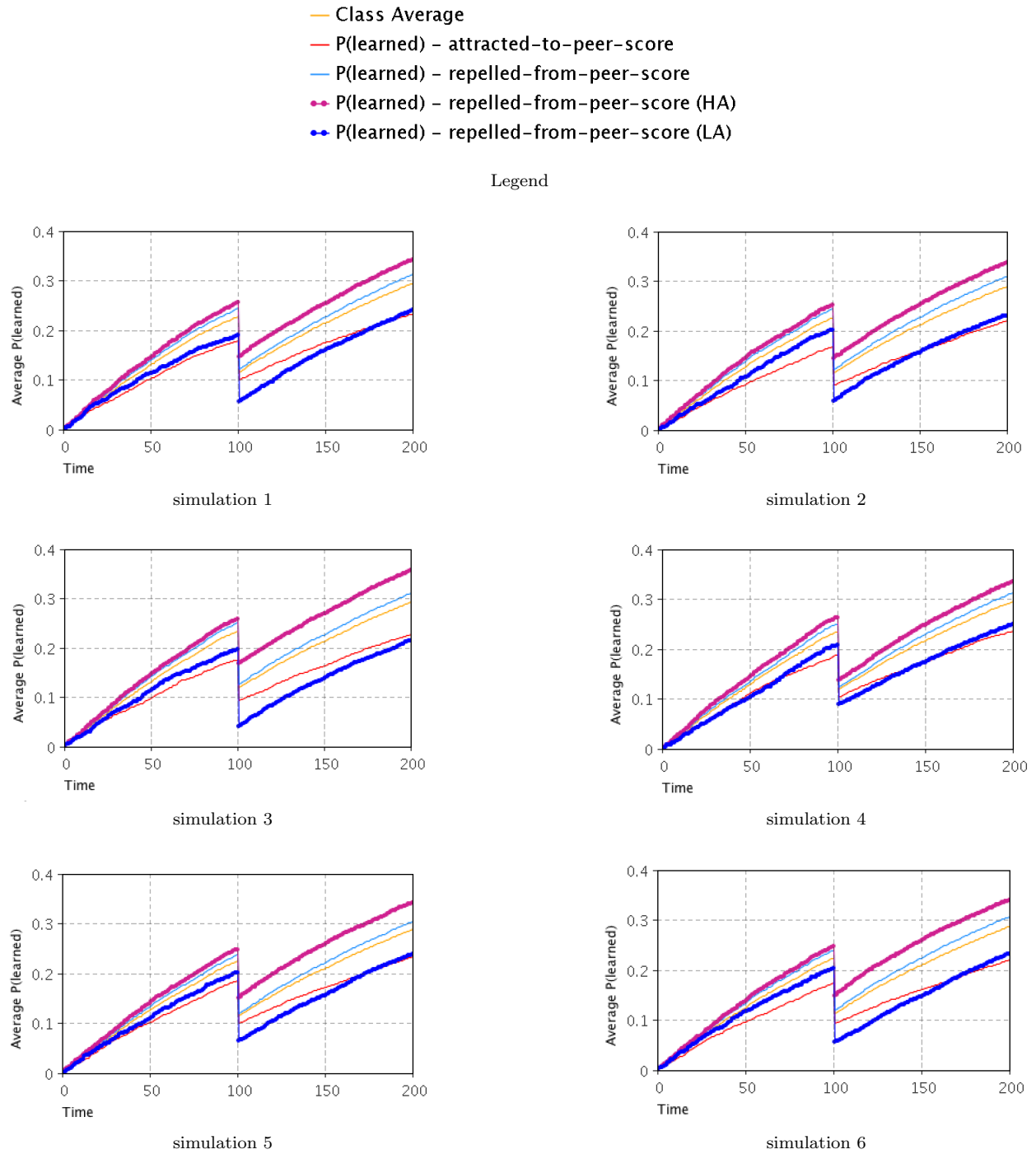


Figure A.6: mostlyrepelled, easy mode

APPENDIX B

CFLS PLANNER SIMULATIONS

This appendix shows further details of the experiment described in Chapter 4 for the very interested reader. The CFLS planner was run 25 times using all pairings of the values of b , f and s ranging from 1 to 5 where $f = s$. The meanings of b , f and s are described in Section 4.2. Each simulation produced two measurements (score on Final Exam and % LOs Mastered), which are described in Section 4.3. The graphs below show these two measurements for each of the 25 simulations. For compactness, two graphs are printed per line.

In all cases, the x-axis is time. In graphs on the left, the y-axis shows the score on Final Exam. In graphs on the right, the y-axis shows the % LOs Mastered. All measurements taken at time=200 are summarized in Figs. 4.2 and 4.3, broken down by aptitude group (low, medium and high). The values in Figs. 4.2 and 4.3 can be compared with the baseline measurements shown in Chapter 4 in Table 4.1.

By looking at the graphs below, in some cases it's easy to see the effects of different values of b , f and s . For example, the low aptitude learners (orange) are clearly achieving higher scores in Fig. B.1 than in Fig. B.3. In other cases it is less obvious, so t-tests were used to compare the individual data points between different simulation runs to determine. The t-test results are shown in Appendix C.

b=1, f=1, s=1

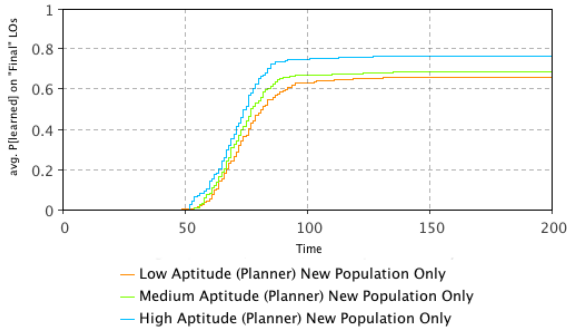


Figure B.1: score on Final Exam

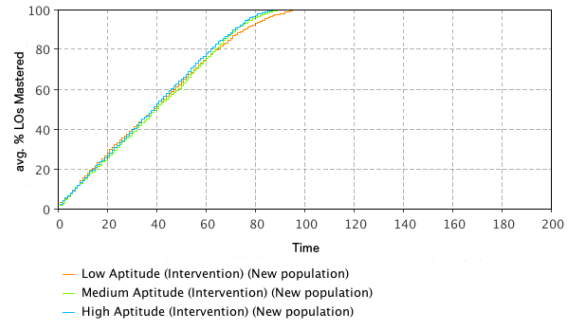


Figure B.2: % LOs Mastered

b=1, f=2, s=2

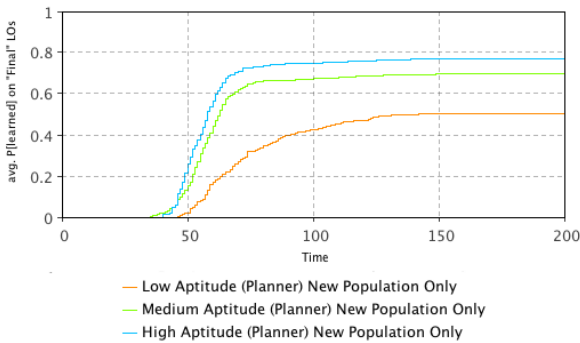


Figure B.3: score on Final Exam

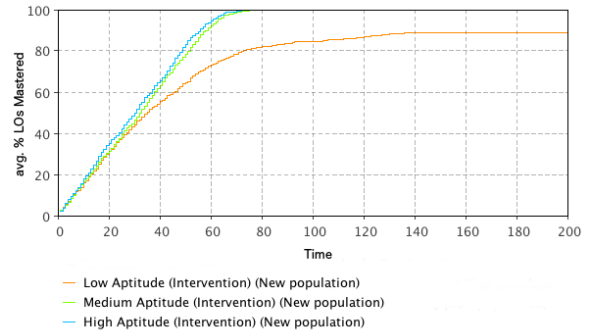


Figure B.4: % LOs Mastered

b=1, f=3, s=3

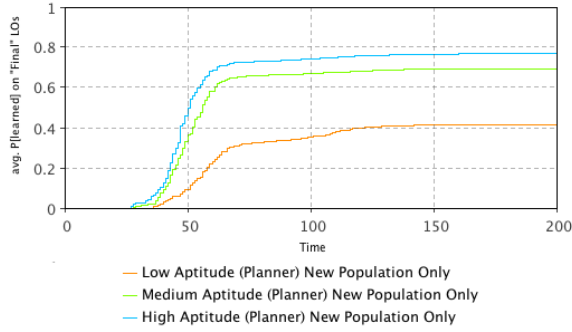


Figure B.5: score on Final Exam

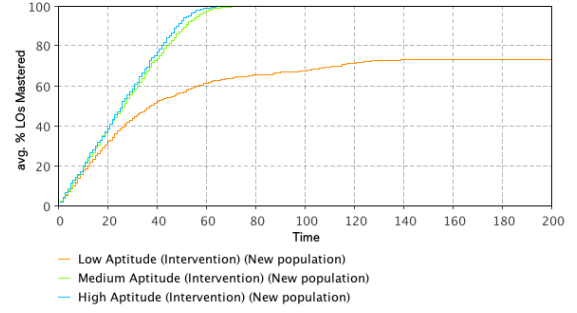


Figure B.6: % LOs Mastered

b=1, f=4, s=4

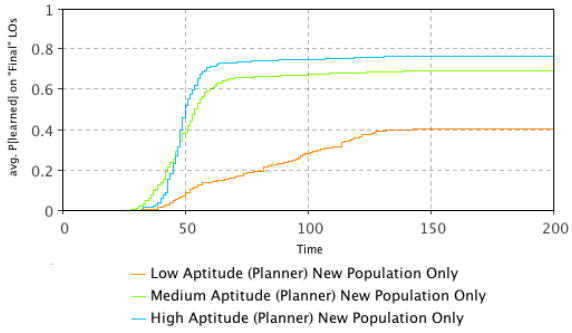


Figure B.7: score on Final Exam

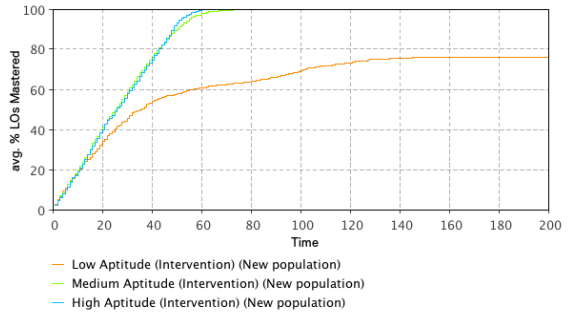


Figure B.8: % LOs Mastered

b=1, f=5, s=5

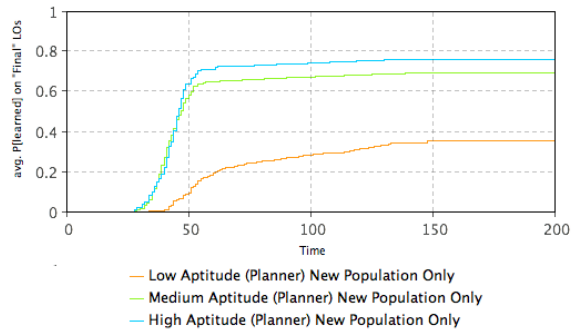


Figure B.9: score on Final Exam

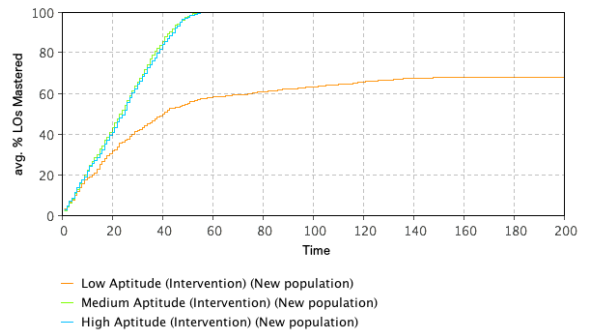


Figure B.10: % LOs Mastered

b=2, f=1, s=1

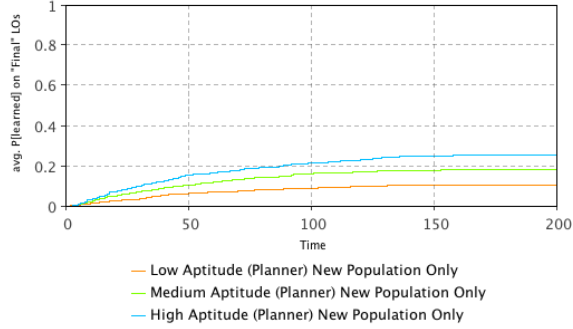


Figure B.11: score on Final Exam

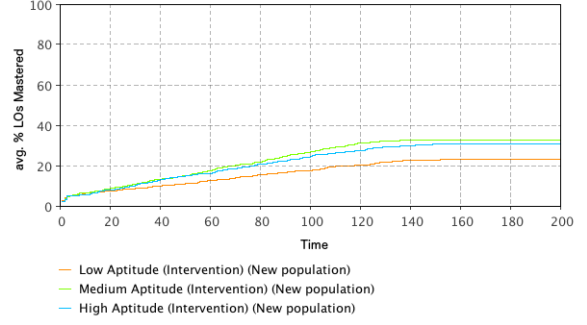


Figure B.12: % LOs Mastered

b=2, f=2, s=2

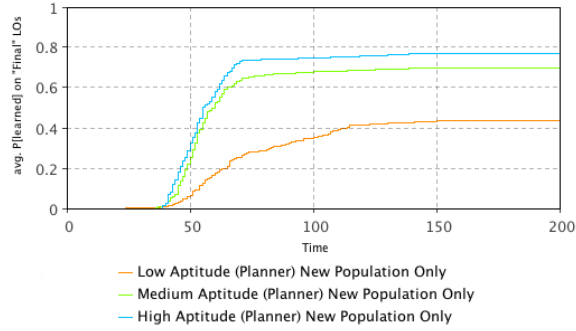


Figure B.13: score on Final Exam

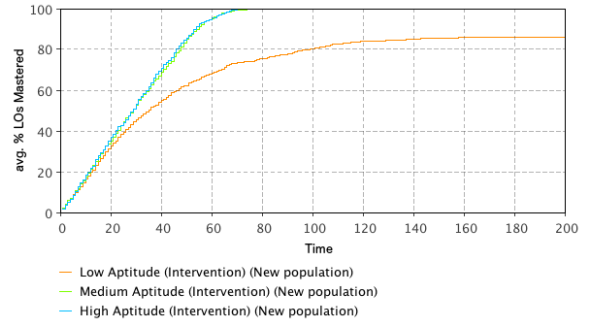


Figure B.14: % LOs Mastered

b=2, f=3, s=3

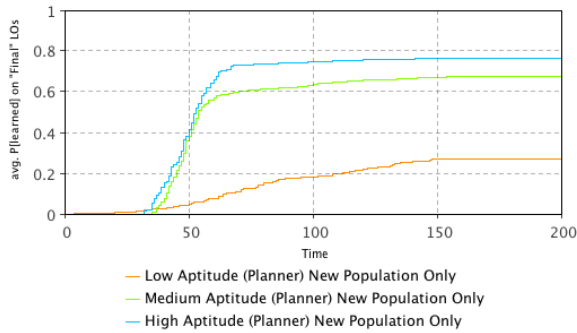


Figure B.15: score on Final Exam

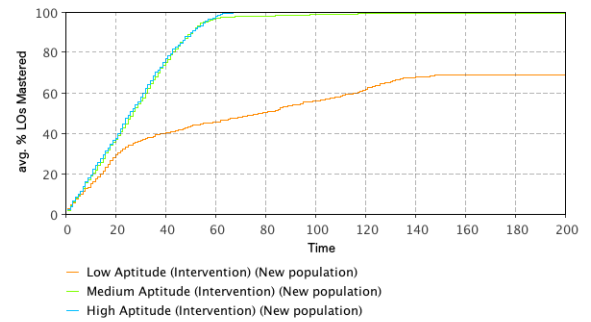


Figure B.16: % LOs Mastered

$b=2, f=4, s=4$

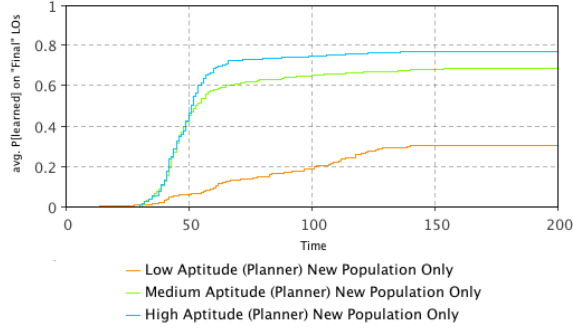


Figure B.17: score on Final Exam

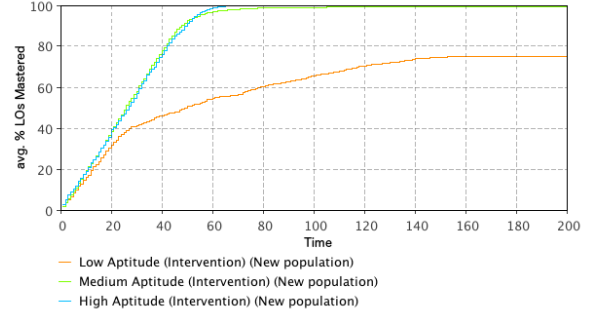


Figure B.18: % LOs Mastered

$b=2, f=5, s=5$

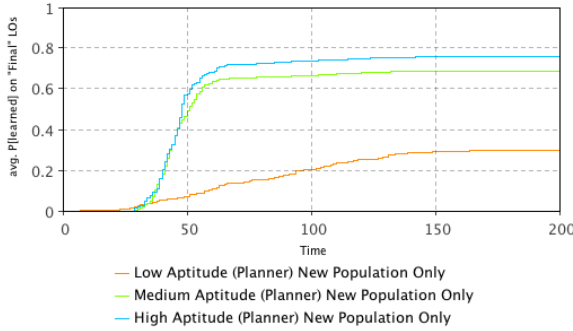


Figure B.19: score on Final Exam

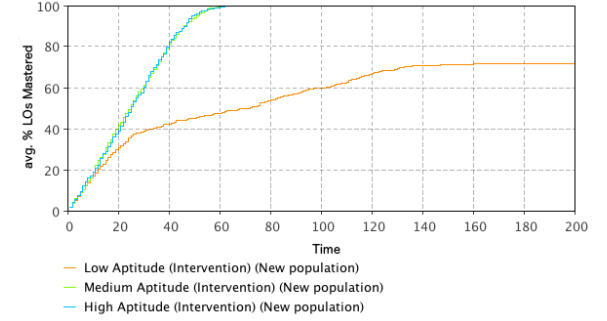


Figure B.20: % LOs Mastered

$b=3, f=1, s=1$

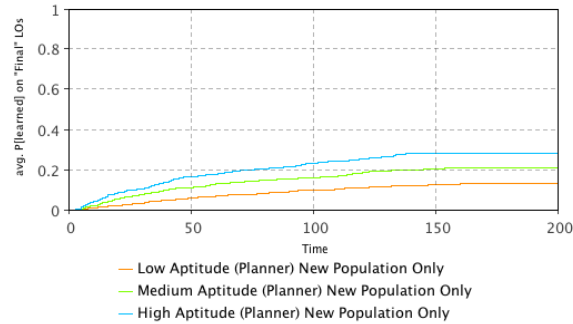


Figure B.21: score on Final Exam

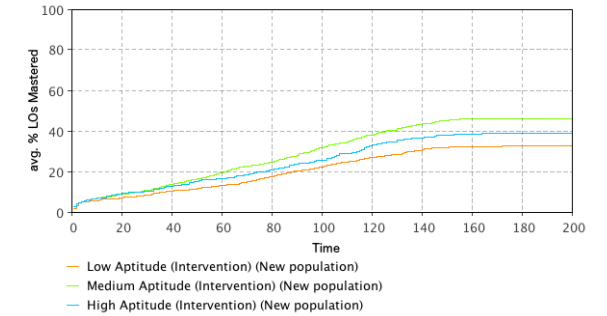


Figure B.22: % LOs Mastered

b=3, f=2, s=2

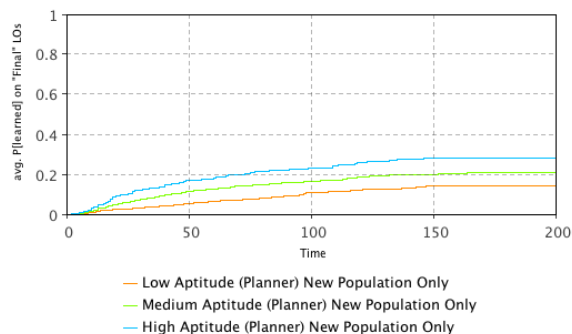


Figure B.23: score on Final Exam

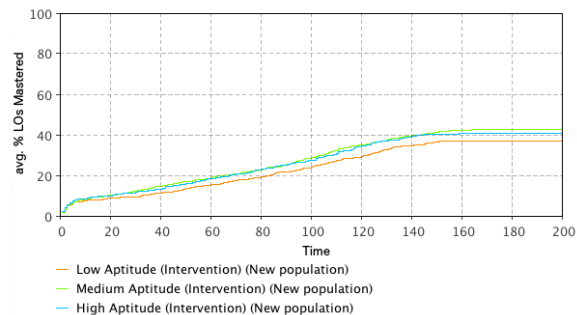


Figure B.24: % LOs Mastered

b=3, f=3, s=3

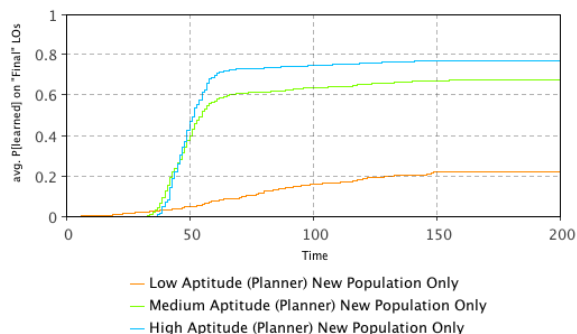


Figure B.25: score on Final Exam

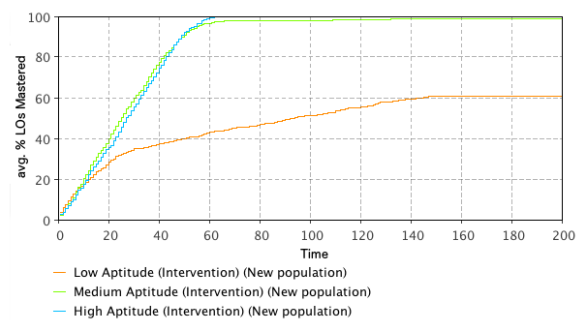


Figure B.26: % LOs Mastered

b=3, f=4, s=4

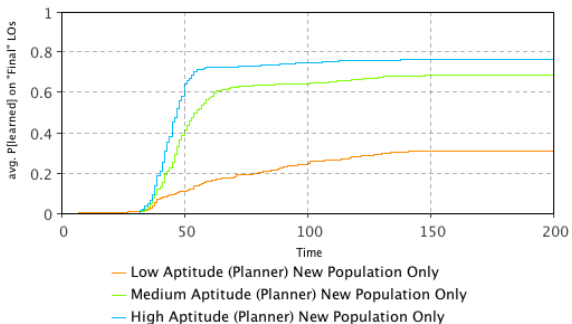


Figure B.27: score on Final Exam

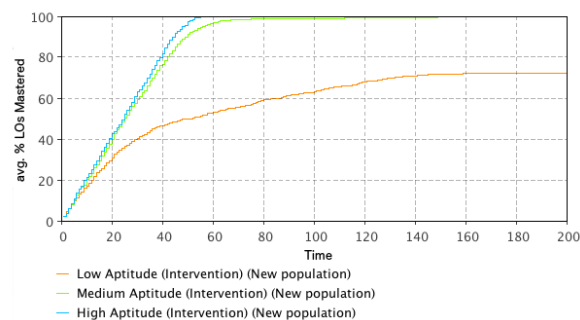


Figure B.28: % LOs Mastered

b=3, f=5, s=5

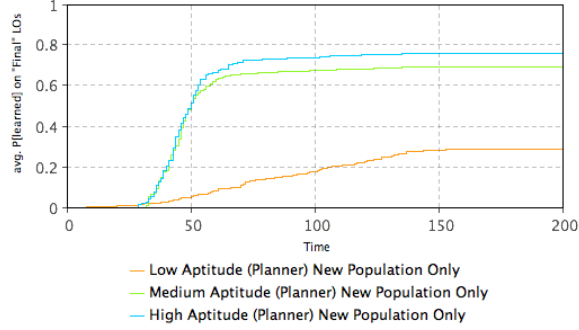


Figure B.29: score on Final Exam

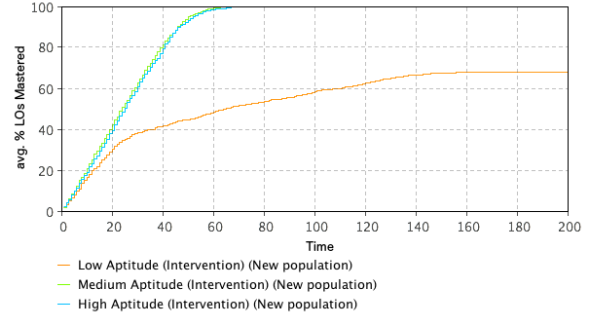


Figure B.30: % LOs Mastered

b=4, f=1, s=1

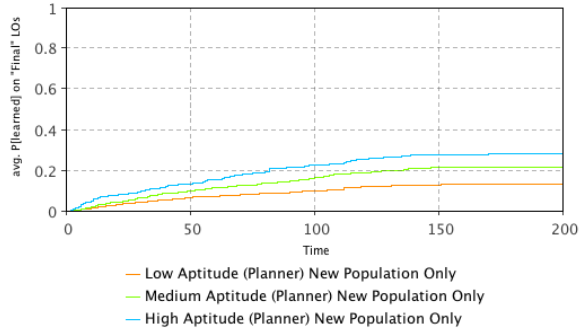


Figure B.31: score on Final Exam

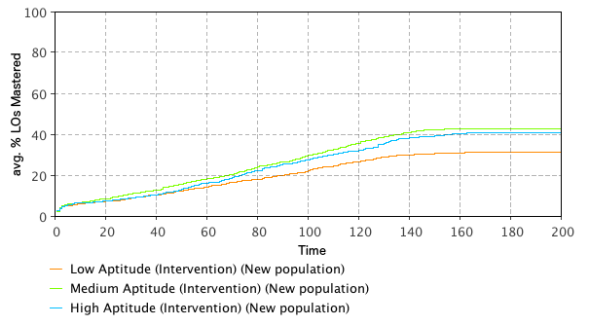


Figure B.32: % LOs Mastered

b=4, f=2, s=2

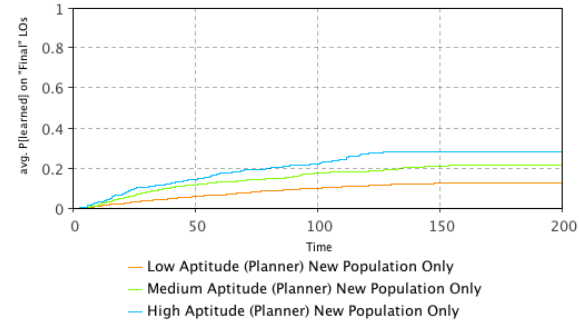


Figure B.33: score on Final Exam

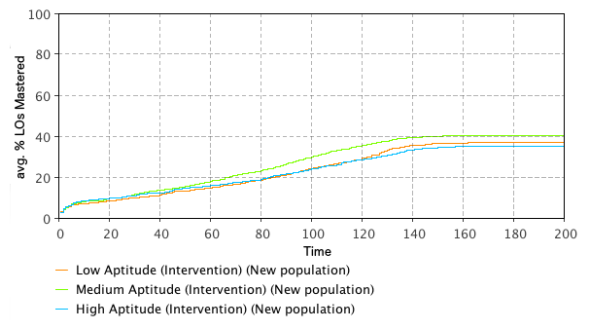


Figure B.34: % LOs Mastered

$b=4, f=3, s=3$

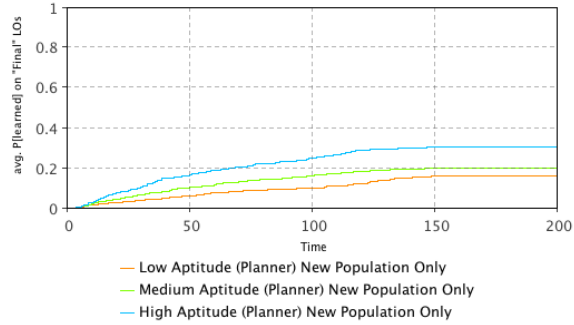


Figure B.35: score on Final Exam

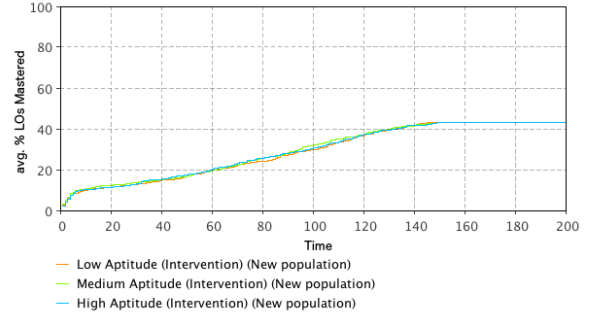


Figure B.36: % LOs Mastered

$b=4, f=4, s=4$

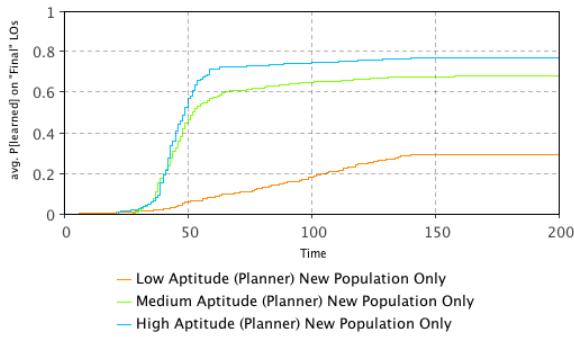


Figure B.37: score on Final Exam

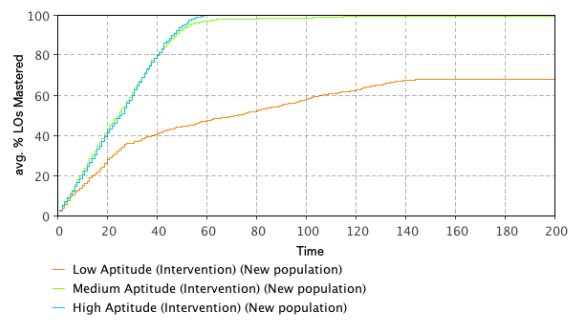


Figure B.38: % LOs Mastered

$b=4, f=5, s=5$

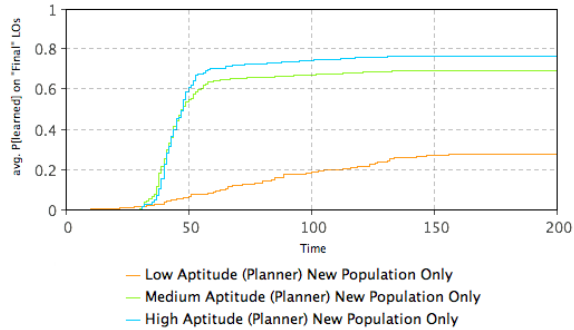


Figure B.39: score on Final Exam

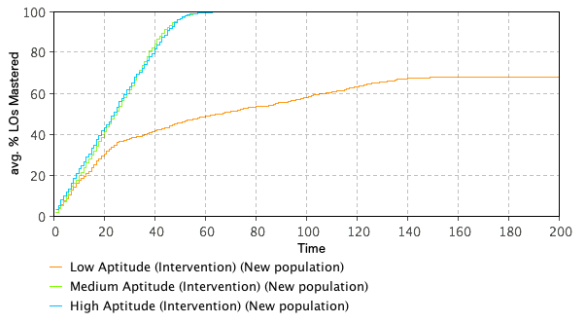


Figure B.40: % LOs Mastered

$b=5, f=1, s=1$

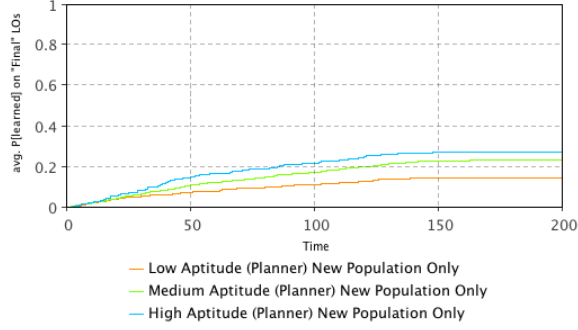


Figure B.41: score on Final Exam

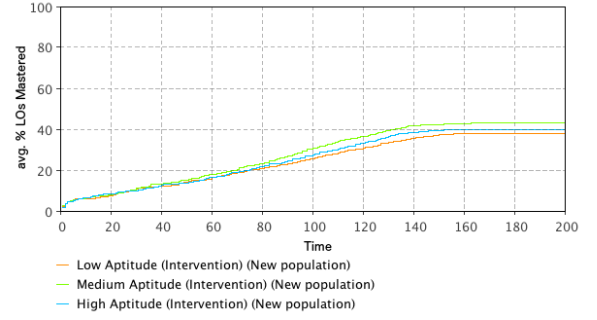


Figure B.42: % LOs Mastered

$b=5, f=2, s=2$

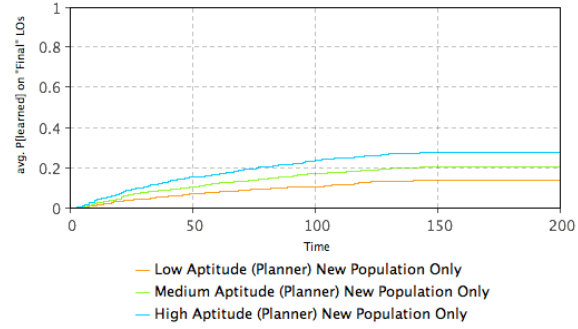


Figure B.43: score on Final Exam

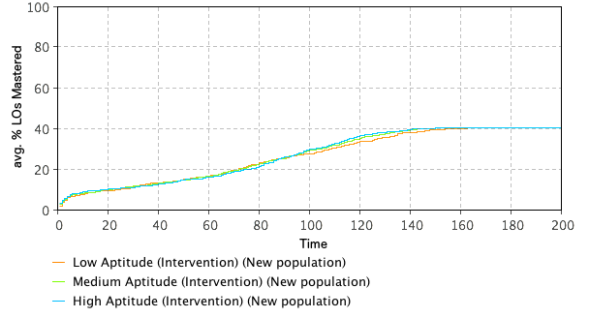


Figure B.44: % LOs Mastered

$b=5, f=3, s=3$

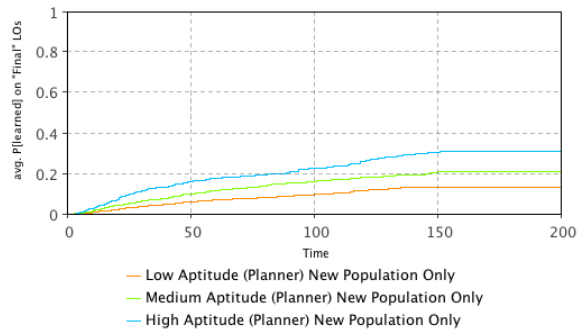


Figure B.45: score on Final Exam

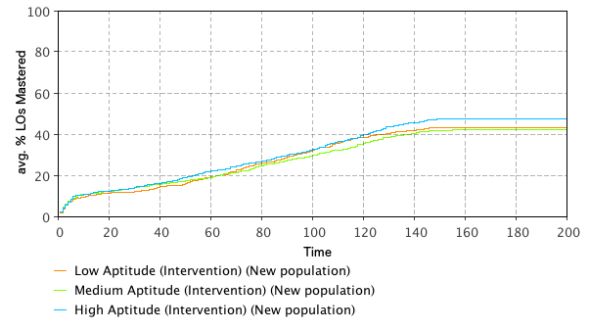


Figure B.46: % LOs Mastered

$b=5, f=4, s=4$

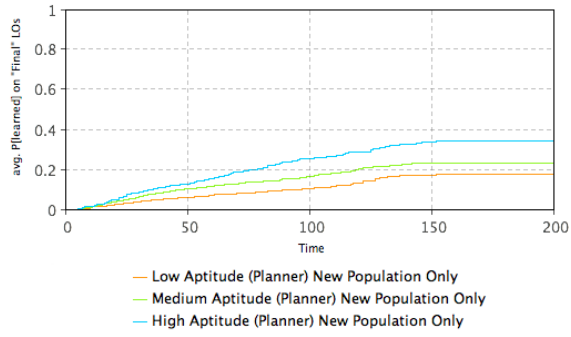


Figure B.47: score on Final Exam

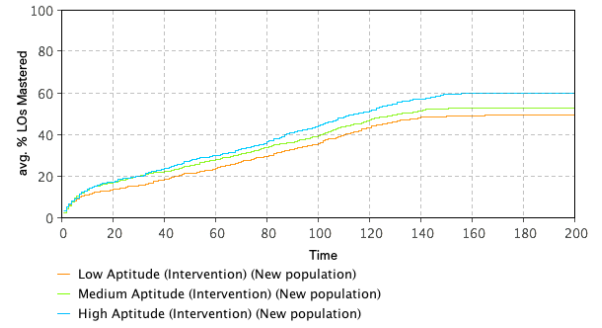


Figure B.48: % LOs Mastered

$b=5, f=5, s=5$

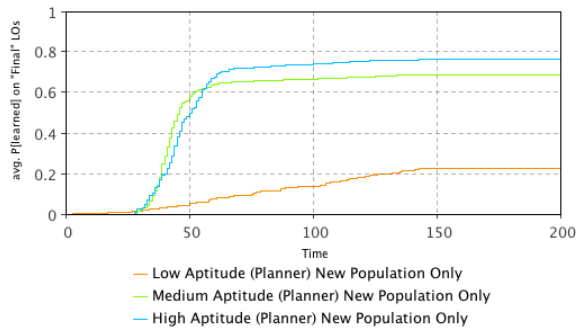


Figure B.49: score on Final Exam

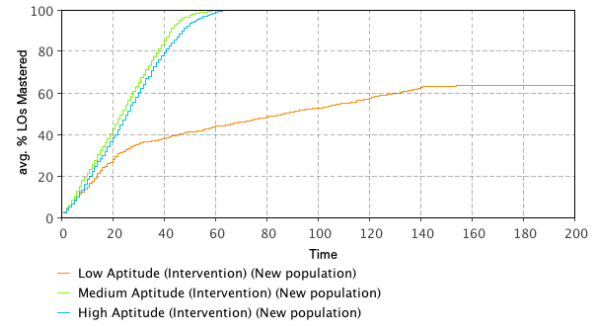


Figure B.50: % LOs Mastered

APPENDIX C

COMPARISON OF CFLS PLANNER SIMULATIONS

Student's t-test was used to check for significant differences in changes to b and $f = s$. Paired t-tests can be used because the simulated learners have exactly the same characteristics in all the simulation runs. For example, learner 3 always has **aptitude** = 4, so, there is no difference in that learner between simulation runs. A two-tailed t-test was used because it was not certain whether one distribution was going to be higher or lower than the other.

Columns are labelled with abbreviations. For example, 111 means $b = 1, f = s = 1$. Similarly, 211 means $b = 2, f = s = 1$. A t-test between 111 and 211 is a check whether there was a significant difference by increasing b from 1 to 2 while holding f and s the same. This is represented in the first cell of the table, 111 vs 211. The t-tests were calculated separately per aptitude group in order to check, for example, whether changing from 111 to 211 might have a different impact on the high aptitude group than on the low aptitude group. The table rows are shaded using the same colour scheme as Appendix B (high aptitude = blue, medium aptitude = green, low aptitude = orange).

In the tables below, significant values ($p < 0.05$) are **bolded**. The hypothesis is that there is no difference caused by changing the values of b, f or s . For instance, for 111 vs 211, the hypothesis is that by changing b from 1 to 2, there will be no impact on the learners' score on Final Exam nor on % LOs Mastered. When p-values turn out to be significant, it is concluded that indeed there was an impact by making this change to b . Cells marked n/a represent cases where 100% of the LOs had been mastered by all simulated students; because the distributions are identical, the t-test doesn't apply.

Each column, for example '111 vs 211' is labelled with a symbol. \blacktriangle means that the two columns being compared are within the blue triangle of failure. \blacktriangle means that the two columns being compared are within the red success triangle. \blacksquare means that the two columns fall along the border of the red and blue triangles.

C.1 Changes in b

Table C.1: P-values for $b = \{1...5\}, f = 1, s = 1$

Student's t-test	111 vs 211 \blacksquare	211 vs 311 \blacktriangle	311 vs 411 \blacktriangle	411 vs 511 \blacktriangle
HIGH APTITUDE	N = 18	N = 18	N = 18	N = 18
p-value (Score on Final)	3.83×10^{-14}	0.2880	0.7905	0.4450
p-value (% LOs Mastered)	2.99×10^{-12}	0.1164	0.9311	0.7586
MEDIUM APTITUDE	N = 26	N = 26	N = 26	N = 26
p-value (Score on Final)	7.87×10^{-26}	0.1571	0.9758	0.1881
p-value (% LOs Mastered)	3.45×10^{-17}	0.0083	0.2449	0.5862
LOW APTITUDE	N = 21	N = 21	N = 21	N = 21
p-value (Score on Final)	1.36×10^{-23}	0.0627	0.8131	0.2835
p-value (% LOs Mastered)	1.23×10^{-16}	0.0285	0.6955	0.1322

Table C.2: P-values for $b = \{1...5\}$, $f = 2$ $s = 2$

Student's t-test	122 vs 222 ▲	222 vs 322 ■ ■	322 vs 422 ▲	422 vs 522 ▲
HIGH APTITUDE	N = 18	N = 18	N = 18	N = 18
p-value (Score on Final)	0.8865	1.49×10^{-19}	0.6656	0.9498
p-value (% LOs Mastered)	n/a	4.30×10^{-15}	0.0372	0.0710
MEDIUM APTITUDE	N = 26	N = 26	N = 26	N = 26
p-value (Score on Final)	0.6416	1.72×10^{-26}	0.4707	0.2589
p-value (% LOs Mastered)	n/a	3.66×10^{-16}	0.5219	0.5579
LOW APTITUDE	N = 21	N = 21	N = 21	N = 21
p-value (Score on Final)	0.0909	3.41×10^{-6}	0.1021	0.3038
p-value (% LOs Mastered)	0.3046	1.38×10^{-10}	0.8755	0.2857

Table C.3: P-values for $b = \{1...5\}$, $f = 3$ $s = 3$

Student's t-test	133 vs 233 ▲	233 vs 333 ▲	333 vs 433 ■ ■	433 vs 533 ▲
HIGH APTITUDE	N = 18	N = 18	N = 18	N = 18
p-value (Score on Final)	0.7296	0.0172	5.01×10^{-13}	0.8021
p-value (% LOs Mastered)	n/a	n/a	1.95×10^{-13}	0.1498
MEDIUM APTITUDE	N = 26	N = 26	N = 26	N = 26
p-value (Score on Final)	0.1086	0.6294	1.29×10^{-22}	0.0593
p-value (% LOs Mastered)	0.0309	0.0169	3.35×10^{-17}	0.8851
LOW APTITUDE	N = 21	N = 21	N = 21	N = 21
p-value (Score on Final)	0.0017	0.2648	0.0322	0.0285
p-value (% LOs Mastered)	0.3856	0.0701	0.0029	0.9758

Table C.4: P-values for $b = \{1...5\}$, $f = 4$ $s = 4$

Student's t-test	144 vs 244 ▲	244 vs 344 ▲	344 vs 444 ▲	444 vs 544 ■ ■
HIGH APTITUDE	N = 18	N = 18	N = 18	N = 18
p-value (Score on Final)	0.7777	0.3101	0.2835	9.74×10^{-12}
p-value (% LOs Mastered)	n/a	n/a	n/a	4.35×10^{-9}
MEDIUM APTITUDE	N = 26	N = 26	N = 26	N = 26
p-value (Score on Final)	0.3306	0.9136	0.8117	2.70×10^{-19}
p-value (% LOs Mastered)	0.0697	0.5739	0.7876	3.03×10^{-15}
LOW APTITUDE	N = 21	N = 21	N = 21	N = 21
p-value (Score on Final)	0.0349	0.9150	0.4767	0.0106
p-value (% LOs Mastered)	0.4899	0.3903	0.0093	0.0012

Table C.5: P-values for $b = \{1...5\}$, $f = 5$ $s = 5$

Student's t-test	155 vs 255 ▲	255 vs 355 ▲	355 vs 455 ▲	455 vs 555 ▲
HIGH APTITUDE	N = 18	N = 18	N = 18	N = 18
p-value (Score on Final)	0.8537	0.7079	0.3568	0.9014
p-value (% LOs Mastered)	n/a	n/a	n/a	n/a
MEDIUM APTITUDE	N = 26	N = 26	N = 26	N = 26
p-value (Score on Final)	0.5621	0.3194	0.4425	0.2073
p-value (% LOs Mastered)	n/a	n/a	n/a	n/a
LOW APTITUDE	N = 21	N = 21	N = 21	N = 21
p-value (Score on Final)	0.0474	0.9414	0.5609	0.1786
p-value (% LOs Mastered)	0.3441	0.2196	0.9702	0.2882

C.2 Changes in f

Table C.6: P-values for $b = 1$, f and $s = \{1...5\}$

Student's t-test	111 vs 122 ▲	122 vs 133 ▲	133 vs 144 ▲	144 vs 155 ▲
HIGH APTITUDE	N = 18	N = 18	N = 18	N = 18
p-value (Score on Final)	0.3186	0.3722	0.5439	0.1762
p-value (% LOs Mastered)	n/a	n/a	n/a	n/a
MEDIUM APTITUDE	N = 26	N = 26	N = 26	N = 26
p-value (Score on Final)	0.0085	0.2174	0.8150	0.6163
p-value (% LOs Mastered)	n/a	n/a	n/a	n/a
LOW APTITUDE	N = 21	N = 21	N = 21	N = 21
p-value (Score on Final)	0.0010	0.0835	0.8638	0.2153
p-value (% LOs Mastered)	0.0012	0.0253	0.3077	0.0152

Table C.7: P-values for $b = 2$, f and $s = \{1...5\}$

Student's t-test	211 vs 222 ■ ■	222 vs 233 ▲	233 vs 244 ▲	244 vs 255 ▲
HIGH APTITUDE	N = 18	N = 18	N = 18	N = 18
p-value (Score on Final)	4.30×10^{-14}	0.2872	0.2219	0.1065
p-value (% LOs Mastered)	2.99×10^{-12}	n/a	n/a	n/a
MEDIUM APTITUDE	N = 26	N = 26	N = 26	N = 26
p-value (Score on Final)	2.31×10^{-25}	0.0536	0.2508	0.8103
p-value (% LOs Mastered)	3.45×10^{-17}	0.0309	0.6636	0.0697
LOW APTITUDE	N = 21	N = 21	N = 21	N = 21
p-value (Score on Final)	1.86×10^{-6}	0.0025	0.3814	0.7983
p-value (% LOs Mastered)	5.83×10^{-12}	0.0001	0.1123	0.2687

Table C.8: P-values for $b = 3$, f and $s = \{1...5\}$

Student's t-test	311 vs 322 ▲	322 vs 333 ■ ■	333 vs 344 ▲	344 vs 355 ▲
HIGH APTITUDE	N = 18	N = 18	N = 18	N = 18
p-value (Score on Final)	0.4762	2.56×10^{-21}	0.1026	0.3739
p-value (% LOs Mastered)	0.5799	4.30×10^{-15}	n/a	n/a
MEDIUM APTITUDE	N = 26	N = 26	N = 26	N = 26
p-value (Score on Final)	0.6404	1.68×10^{-22}	0.1588	0.5483
p-value (% LOs Mastered)	0.1483	2.58×10^{-16}	0.0152	0.0222
LOW APTITUDE	N = 21	N = 21	N = 21	N = 21
p-value (Score on Final)	0.5547	0.0156	0.0231	0.6565
p-value (% LOs Mastered)	0.3041	0.0003	0.0004	0.0419

Table C.9: P-values for $b = 4$, f and $s = \{1...5\}$

Student's t-test	411 vs 422 ▲	422 vs 433 ▲	433 vs 444 ■ ■	444 vs 455 ▲
HIGH APTITUDE	N = 18	N = 18	N = 18	N = 18
p-value (Score on Final)	0.5865	0.2678	1.19×10^{-12}	0.2952
p-value (% LOs Mastered)	0.2380	0.0371	1.95×10^{-13}	n/a
MEDIUM APTITUDE	N = 26	N = 26	N = 26	N = 26
p-value (Score on Final)	0.6823	0.1272	3.27×10^{-27}	0.2264
p-value (% LOs Mastered)	0.6474	0.5720	5.77×10^{-17}	0.1101
LOW APTITUDE	N = 21	N = 21	N = 21	N = 21
p-value (Score on Final)	0.6752	0.0065	0.0001	0.8330
p-value (% LOs Mastered)	0.1820	0.0761	0.0004	0.7050

Table C.10: P-values for $b = 5$, f and $s = \{1...5\}$

Student's t-test	511 vs 522 ▲	522 vs 533 ▲	533 vs 544 ▲	544 vs 555 ■ ■
HIGH APTITUDE	N = 18	N = 18	N = 18	N = 18
p-value (Score on Final)	0.6989	0.1365	0.3215	1.08×10^{-11}
p-value (% LOs Mastered)	0.7906	0.0346	0.0224	4.35×10^{-9}
MEDIUM APTITUDE	N = 26	N = 26	N = 26	N = 26
p-value (Score on Final)	0.0130	0.2066	0.4924	9.39×10^{-20}
p-value (% LOs Mastered)	0.1645	0.2852	0.0890	3.58×10^{-15}
LOW APTITUDE	N = 21	N = 21	N = 21	N = 21
p-value (Score on Final)	0.5588	0.7483	0.0854	0.1070
p-value (% LOs Mastered)	0.4377	0.4911	0.1764	0.0023

APPENDIX D

PREREQUISITE GRAPH USED IN CHAPTER 4

Each circle represents a learning object. Arrows indicate prerequisite relationships. For example, LO_6 is a prerequisite to LO_2 . The shaded LOs indicate the select few advanced LOs used for the Final Exam.

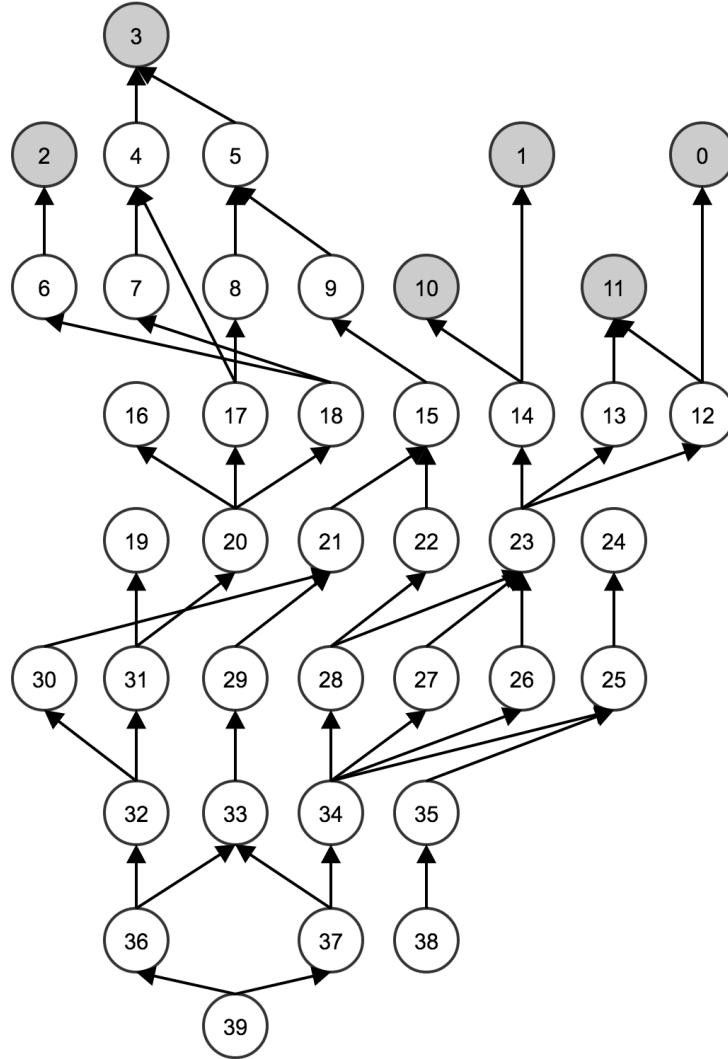


Figure D.1: Prerequisite Graph used in Chapter 4